

Quick & Easy Way

`$obj -> bd;`

অ্যাডভান্সড পিএইচপি এবং মাইএসকিউএস
Your Professional Trainer...


অবজেক্ট ওরিয়েন্টেড প্রোগ্রামিং পিএইচপি

From Beginner to Advanced

Concepts | Techniques | Code

বর্তমান অবস্থা


বর্তমান অবস্থা



পিএইচপি ডেভেলপারদের জন্য এবং পিএইচপি ডিভিক ফ্রেন্ডসের জন্য ও সিএমএস গুলো শেখার জন্য অবজেক্ট ওরিয়েন্টেড পিএইচপি অত্যাবশ্যিক।

সম্পাদনায় :
বুকেরিডিং

ডাউনলোড করুন
bookbd.info



অবজেক্ট অরিয়েন্টেড পিএইচপি

অবজেক্ট অরিয়েন্টেড পিএইচপি

সম্পাদনায় : বুকবিডি সিরিজ এবং অন্যান্য প্রফেশনালগণ।

স্বত্ব : বুকবিডি সিরিজ

প্রকাশক : শরীফ হাসান তরফদার

জ্ঞানকোষ প্রকাশনী

৩৮/২-ক বাংলাবাজার, ঢাকা-১১০০।

ফোন- ৭১১৮৪৪৩, ৮৬২৩২৫১, ৮১১২৪৪১

E-mail : gyankoshprokashoni@gmail.com
gk_tarafder@yahoo.com

প্রকাশকাল : মে ২০১১ ইংরেজি

প্রচ্ছদ : আর্টিস্ট ক্রিয়েটিভ মিডিয়া

কম্পোজ : কমপিউটার লিটারেসি হাউস

মেকাপ : বিশ্বজিৎ দাস

মুদ্রণ : নোভা প্রেস এন্ড পাবলিকেশন্স

১৫/বি, মিরপুর রোড ঢাকা-১২০৫।

ফোন : ৯৬৬৭৯১৯

ISBN : 978-

মূল্য : ৩৭০ টাকা মাত্র। (সিডি সহ)

কৃতজ্ঞতা

ধন্যবাদ :

১. মোঃ শরিফ হোসেন (শিহাব)
২. মোঃ কামাল হোসেন (মামুন)
৩. মোঃ ইমরান হোসেন
৪. তানভীর আহম্মেদ
৫. সেকান্দার বাদশাহ্
৬. আবু সাঈদ ইয়াসিন

অবজেক্ট অরিয়েন্টেড পিএইচপি

এই বইটির সাথে ফ্রি যা রয়েছে :



ফ্রি সিডি

অবজেক্ট অরিয়েন্টেড পিএইচপি

প্রিয় পাঠক,

আমরা বইটি লেখার সময় আপ্রাণ চেষ্টা করেছি, একটি ভাল বই লেখার জন্য। তারপরও আমাদের অজান্তে হয়তো বা কোন ভুল থাকতে পারে। সেক্ষেত্রে উক্ত ভুল-ত্রুটি গুলো ক্ষমার দৃষ্টিতে দেখার অনুরোধ রইল। এবং আপনাদের মতামত আমাদেরকে ই-মেইলের মাধ্যমে জানালে উপকৃত হবো। আমাদের ই-মেইল ঠিকানা : infobook7@gmail.com।

বুকবিডি সিরিজের প্রতিটি বই লেখা হয় প্রফেশনালদের বাস্তব অভিজ্ঞতার আলোকে। তথ্যপ্রযুক্তিতে দক্ষ জনশক্তি তৈরী করতে প্রতিটি বই হবে একজন প্রশিক্ষক। এ লক্ষ্য সামনে রেখে বুকবিডি সিরিজের পদ যাত্রা। যার মূল লক্ষ্য হচ্ছে বাংলা ভাষায় তথ্য প্রযুক্তির বিস্তার ঘটানো। আমাদের বইয়ের ফ্রি অংশ www.bookbd.info থেকে ডাউনলোড করা যাবে।

লেখকের কথা

> মহান আল্লাহ তা'আলার অশেষ শুকরিয়া, বাংলাদেশে সর্ব প্রথম বাংলা ভাষায় “অবজেক্ট অরিয়েন্টেড পিএইচপি” বইটি পাঠকদের হাতে তুলে দেওয়ার জন্য। দেশের অগণিত পাঠকদের চাহিদা আর অনুরোধই আমাকে এই বইটি লেখার অনুপ্রেরনা যুগিয়েছে।

> বর্তমানে যতগুলো প্রোগ্রামিং ল্যাংগুয়েজ রয়েছে তার মধ্য অন্যতম হচ্ছে অবজেক্ট অরিয়েন্টেড পিএইচপি। এটি অবজেক্ট গুলোকে খুব সহজে মেনটেইন করে। সফটওয়্যার অবজেক্ট মডেল এর সাথে বাস্তব জীবনের অবজেক্ট মডেলের অনেক মিল রয়েছে। পিএইচপি কে হাই লেভেলিং স্ক্রিপ্টিং ল্যাংগুয়েজ ও বলা হয়। যেটি দিয়ে ডাইনামিক ওয়েব সাইট। ওয়েব বেসড এ্যাপ্লিকেশন তৈরী করা অনেক সহজ। এটি একটি ওপেনসোস ল্যাংগুয়েজ (সার্ভার সাইড স্ক্রিপ্টিং ল্যাংগুয়েজ)।

> বর্তমান বিশ্বের সাথে তাল মিলিয়ে চলার জন্য অথবা ক্যারিয়ারে নিজের এশটি অবস্থান তৈরী করার জন্য মানুষ আজ যে কোন একটি বিষয়ের উপর দক্ষ হচ্ছে। সেক্ষেত্রে ভবিষ্যৎ এর কথা চিন্তা করে অনেকেই কম্পিউটার প্রোগ্রামিং ল্যাংগুয়েজ পিএইচপিকে বেছে নেয়। এর অন্যতম কারন হচ্ছে ওয়েব ভিত্তিক সফটওয়্যার তৈরিতে অবজেক্ট অরিয়েন্টেড পিএইচপি একটি শক্তিশালী টুলস। যেহেতু দিন দিন ওয়েব সাইট তৈরি তথা ওয়েব ডেভেলপিং এর চাহিদা বৃদ্ধি পাচ্ছে। সেহেতু বলা যায় একজন ওয়েব ডেভেলপিং হয়ে একটি উন্নত ক্যারিয়ার তৈরি করা সম্ভব। এখন চাকুরীর জন্য আর বিভিন্ন কোম্পানীর কাছে ঘুরে বেড়াতে হয় না। আপনি চাইলেই ঘরে বসে নিজের কর্মসংস্থান সৃষ্টি করতে পারেন। বাংলাদেশ এবং বহির্বিশ্বের বেশির ভাগ কোম্পানী পিএইচপি নিয়ে কাজ করে। সে জন্য পিএইচপি ডেভেলপারদের চাহিদা অনেক বেশি।

> বইটিতে অবজেক্ট অরিয়েন্টেড কনসেপ্ট নিয়ে বিস্তারিত আলোচনা করা হয়েছে। যেমনঃ ইনহেরিটেন্স, পলিমরফিজম, ইন্টারফেস, অ্যাবস্ট্রাক, এনক্যাপসুলেশন। এছাড়াও অবজেক্ট এবং ক্লাস, ভ্যারিয়েবল, কনস্ট্রাক্ট, কী-ওয়ার্ড, অপারেটরস, এক্সপ্রেশন, কন্ট্রোল স্ট্রাকচার, লুপিং, ব্রেক কনটিনিউটি, অ্যারে, ফাংশন, মেথড, মডিফায়ার এবং কী-ওয়ার্ড ইত্যাদি। অবজেক্ট অরিয়েন্টেড পিএইচপি এবং মাইসিকিউএল ডেটাবেজ ব্যবহার করে প্রজেক্ট দেওয়া আছে।

> বইটিতে প্রোগ্রামিং ল্যাংগুয়েজ সহজ ভাবে উপস্থাপন করা হয়েছে। বইটি পড়ে অবজেক্ট অরিয়েন্টেড কনসেপ্ট গুলো খুব সহজে বুঝা যাবে। এছাড়াও বাস্তবধর্মী প্রজেক্ট দেওয়া আছে। যেন এটার উপর ভিত্তি করে অন্যান্য এ্যাপ্লিকেশন ডেভেলপ করতে পারে। বইটির লেখায় বা তথ্যগত কোন ভুল ত্রুটি থাকলে ক্ষমা সুন্দর দৃষ্টিতে দেখবেন অথবা আমাদেরকে ই-মেইলের মাধ্যমে জানালে উপকৃত হব। সব শেষে সকলের সাফল্য কামনা করছি।

নিবেদক

বুকবিডি সিরিজ

facebook.com/mmr.sinha

infobook7@gmail.com

www.bookbd.info

facebook.com/ebookbd

পরিবেশক/প্রাপ্তিস্থান

ঢাকাঃ

কাজী ট্রেডার্স
আর্জিজিয়া লাইব্রেরী
সাহিত্য কোষ

খুলনাঃ

সোহাগ বুক সেন্টার
নূর লাইব্রেরী
সালেহিয়া লাইব্রেরী
আর্জিজিয়া লাইব্রেরী

ঢাকাঃ স্টেডিয়াম মার্কেট

অনুপম জ্ঞানভান্ডার

ঢাকাঃ নীলক্ষেত

গোনা বুক সেন্টার
ফরিদা কর্পোরেশন
হক লাইব্রেরী
সাকিব বুক
মডার্ন বুক সেন্টার
পপুলার বুকস্
ফারহান বুকস্
মুক্তা এন্টারপ্রাইজ
ফাহাদ বুকস্

কুষ্টিয়া : বইমেলা, এন এস রোড
চুয়াডাঙ্গা : পুখিঘর লাইব্রেরী, বড় বাজার
যশোর : জনতা লাইব্রেরী, দড়াটানা
বরিশাল : বুকভিলা
কুমিল্লা : শাহীন লাইব্রেরী
বগুড়া :

কোরান হাদীস মঞ্জিল
(উত্তর বঙ্গের একমাত্র পরিবেশক)

ঢাকাঃ নিউমার্কেট

বাংলাদেশ লাইব্রেরী
ঢাকা বুক কর্পোরেশন

ঢাকাঃ আইডিবি ভবন

বুকস্ এন্ড টেক ভিউ

রাজশাহীঃ

আলীগড় লাইব্রেরী
রহমান বুকস্
সূচীপত্র
আব্দুল্লাহ এন্ড ব্রাদার্স
সংলাপ

ঢাকাঃ ফার্মগেট

তোফাজ্জল বুক হাউস
ছাওয়াল বুকস্
সেলিম বুকস্

সিলেটঃ

পপি লাইব্রেরী
নিউ নেশন লাইব্রেরী
বন্ধু লাইব্রেরী

চট্টগ্রামঃ

বুক লাইব্রেরী, আন্দরকিল্লা
সিদ্দিকিয়া লাইব্রেরী, আন্দরকিল্লা
বুক সেন্টার, আন্দরকিল্লা
ন্যাশনাল বুক, আন্দরকিল্লা
প্রাইম বুক, আন্দরকিল্লা
ফ্রেডস বুক, আন্দরকিল্লা
প্রতিভা লাইব্রেরী, আন্দরকিল্লা
আর আলম বুকস্, আন্দরকিল্লা
গুলিস্থান লাইব্রেরী, বিপনী বিতান নিউমার্কেট
অমর বই ঘর, নূপুর মার্কেট, স্টেশন রোড
নিউ বুক ল্যান্ড, চশবাজার
কারেন্ট বুক সেন্টার, জলসা মার্কেট

রংপুরঃ

বিপনী বিচিত্রা
সাহিত্য ভান্ডার
টাউন স্টোর লাইব্রেরী

কলকাতায় পরিবেশক

রিতা ইন্টারন্যাশনাল

৩৬,পি.এন. ব্যানার্জি রোড, কলকাতা

ফোন: ২৫১৩৮৩৫৯, ৯৮৩০৪৩৯৬৭৯

এক নজরে সূচিপত্র

১. অবজেক্ট অরিয়েন্টেড পিএইচপি এর প্রাথমিক আলোচনা	১৪
২. পিএইচপি ও অবজেক্ট অরিয়েন্টেড সম্পর্কে আলোচনা	১৪
৩. ব্রাউজার এবং ওয়েব সার্ভার ইনস্টলেশন সম্পর্কে আলোচনা	১৪
৪. পিএইচপি ফাইল, অবজেক্ট অরিয়েন্টেড পিএইচপি সিনট্যাক্স, পিএইচপি,এইচটিএমএল,ক্রাস, অবজেক্টের আলোচনা	১৪
৫. ভ্যালু, ভ্যারিয়েবল, ভ্যারিয়েবল ডিক্লেয়ার, ডেটা টাইপস, কী-ওয়ার্ড, কনস্ট্যান্ট, কমেন্ট সম্পর্কে আলোচনা	১৪
৬. অপারেটর, এক্সপ্রেশন, অপারেটর টাইপস সম্পর্কে আলোচনা	১৪
৭. স্টেটমেন্ট, কন্ডিশনাল, ইফ, মাল্টিপল ইফ, নেস্টেড ইফ, ইফ এলস, এলস ইফ, সুইচ স্টেটমেন্টের আলোচনা	১৪
৮. লুপের প্রকার ভেদ, ফরলুপ, হোয়াইল লুপ, ডু-হোয়াইল লুপ, ফর ইচ লুপ সম্পর্কে আলোচনা	১৪
৯. ব্রেক, কনটিনিউটি, স্ট্রিং এ্যান্ড কনক্যাটেনেশন	১৪
১০. অ্যারে (Array)	১৪
১১. ফাংশন (Function)	১৪
১২. মেথড (Methods)	১৪
১৩. মডিফায়ার এবং কী-ওয়ার্ড (Modifiers)	১৪
১৪. কনস্ট্রাকটর এবং ডেস্ট্রাকটর (Constructor & Destructor)	১৪
১৫. ইনহেরিটেন্স (Inheritance)	১৪
১৬. পলিমরফিজম (Polymorphism)	১৪
১৭. ইনক্যাপসুলেশন (Encapsulation)	১৪
১৮. অ্যাবস্ট্রাক্ট (Abstract)	১৪
১৯. ইন্টারফেইস (Interface)	১৪
২০. পিএইচপি ফর্ম (ইনপুট এবং আউটপুট)	১৪
২১. পিএইচপি মাইএসকিউএল প্রজেক্ট (input/update/delete)	১৪
২২. টেক্সটভিত্তিক ভ্যালু এবং ডেটাবেইজ	১৪
২৩. রেজিস্ট্রেশন সিস্টেম	১৪

বিস্তারিত সূচিপত্র

১. অবজেক্ট অরিয়েন্টেড পিএইচপি এর প্রাথমিক আলোচনা -----	
১.১ ভূমিকা -----	১৮
১.২ অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং এর সূচনা -----	১৮
১.৩ অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং কি? -----	১৮
১.৪ অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং ল্যাংগুয়েজ সমূহ -----	১৮
১.৫ অবজেক্ট অরিয়েন্টেড এবং প্রোসিডিউর ল্যাংগুয়েজ এর মধ্য পার্থক্য। -----	১৯
১.৬ অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং এর সুবিধা -----	১৯
১.৭ ক্যারিয়ার হিসেবে অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং -----	২০
১.৮ অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং এর জটিলতা -----	২০
১.৯ বহুল ব্যবহৃত অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং শব্দ সমূহ -----	২০
১.১০ এক নজরে অধ্যায় সারাংশ -----	২১
১.১১ প্রশ্ন সমূহ -----	২১
২. পিএইচপি ও অবজেক্ট অরিয়েন্টেড সম্পর্কে আলোচনা -----	
২.১ ভূমিকা -----	২৪
২.২ পিএইচপি কি? -----	২৪
২.৩ পিএইচপি ল্যাংগুয়েজ কিভাবে কাজ করে? -----	২৪
২.৪ ক্যারিয়ার হিসেবে অবজেক্ট অরিয়েন্টেড পিএইচপি -----	২৬
২.৫ অবজেক্ট অরিয়েন্টেড পিএইচপি এর সুবিধা -----	২৬
২.৬ ল্যাম্প -----	২৬
২.৭ অধ্যায় সারাংশ -----	২৭
২.৮ প্রশ্ন -----	২৭
৩. ব্রাউজার এবং ওয়েব সার্ভার ইনস্টলেশন সম্পর্কে আলোচনা -----	২৯
৩.১ ভূমিকা -----	৩০
৩.২ ব্রাউজার ইনস্টলেশন -----	৩০
৩.৩ ওয়েব সার্ভার -----	৩২
৩.৪ ইনস্টল পিএইচপি -----	৩২
৩.৫ ইনস্টল ডেটাবেস মাইএসকিউএল -----	৩৪
৩.৬ এ্যাপাচি (Apache) -----	১৪
৩.৭ আই আই এস (IIS) -----	১৪
৩.৮ জাম্প (XAMPP) -----	১৪
৩.৯ জাম্প ইনস্টল (XAMPP INSTAL) -----	১৪
৩.১০ ওয়াম্প (WAMP) -----	১৪
৩.১১ ওয়াম্প ইনস্টল -----	১৪
৩.১২ রান পিএইচপি প্রোগ্রাম -----	১৪

৪. পিএইচপি ফাইল, অবজেক্ট অরিয়েন্টেড পিএইচপি সিনট্যাক্স, পিএইচপি, এইচটিএমএল, ক্লাস, অবজেক্টের আলোচনা -----	১৪
৪.১ ভূমিকা -----	১৪
৪.২ পিএইচপি ফাইল তৈরী -----	১৪
৪.৩ অবজেক্ট অরিয়েন্টেড পিএইচপি সিনট্যাক্স -----	১৪
৪.৪ ডিসপ্লে Hello Bangladesh -----	১৪
৪.৫ পিএইচপি এবং এইচটিএমএল -----	১৪
৪.৬ ক্লাস তৈরী -----	১৪
৪.৭ অবজেক্ট তৈরী -----	১৪
৪.৮ অবজেক্ট এবং ক্লাস -----	১৪
৫. ভ্যালু, ভ্যারিয়েবল, ভ্যারিয়েবল ডিক্লেয়ার, ডেটা টাইপস, কী-ওয়ার্ড, কনস্ট্যান্ট, কমেন্ট সম্পর্কে আলোচনা -----	১৪
৫.১ ভূমিকা -----	১৪
৫.২ ভ্যালু -----	১৪
৫.৩ ভ্যারিয়েবল -----	১৪
৫.৪ পিএইচপিতে ভ্যারিয়েব ডিক্লেয়ার করার নিয়ম -----	১৪
৫.৫ পিএইচপি ভ্যারিয়েবল -----	১৪
৫.৬ ভ্যারিয়েবল (Variable) ভ্যালু (Value) এবং আউটপুট -----	১৪
৫.৭ পিএইচপিতে ডেটা টাইপস (Data Types) -----	১৪
৫.৮ ভ্যারিয়েবল এ ভ্যালিউ অ্যাসাইন -----	১৪
৫.৯ কী-ওয়ার্ড -----	১৪
৫.১০ কনস্ট্যান্ট (Constant) -----	১৪
৫.১১ কমেন্ট -----	১৪
৫.১২ সংক্ষিপ্ত অধ্যায় -----	১৪
৫.১৩ প্রশ্ন -----	১৪
৬. অপারেটর, এক্সপ্রেশন, অপারেটর টাইপস সম্পর্কে আলোচনা -----	১৪
৬.১ ভূমিকা : -----	১৪
৬.২ অপারেটর (Operator) -----	১৪
৬.৩ এক্সপ্রেশন (Expression) -----	১৪
৬.৪ অপারেটর টাইপস (Operator Types) -----	১৪
৬.৫ অ্যারিথমেটিক অপারেটর (Arithmetic Operators) -----	১৪
৬.৬ অ্যাসাইনমেন্ট অপারেটরস (Assignment Operators) -----	১৪
৬.৭ কম্প্যারিজন অপারেটরস (Comparision Operators) -----	১৪
৬.৮ লজিক্যাল অপারেটর (Logical Operators) -----	১৪
৬.৯ অধ্যায় সারসংক্ষেপঃ -----	১৪
৬.১০ প্রশ্নঃ -----	১৪
৭. স্টেটমেন্ট, কন্ডিশনাল, ইফ, মাল্টিপল ইফ, নেস্টেড ইফ, ইফ এলস, এলস ইফ, সুইচ স্টেটমেন্টের আলোচনা -----	১৪
৭.১ ভূমিকা -----	১৪

৭.২ স্টেটমেন্ট	১৪
৭.৩ কন্ডিশনাল স্টেটমেন্ট	১৪
৭.৪ ইফ স্টেটমেন্ট (If Statement)	১৪
৭.৫ মাল্টিপল ইফ স্টেটমেন্ট (Multiple if Statement)	১৫
৭.৬ নেস্টেড ইফ স্টেটমেন্ট (Nested If)	১৫
৭.৭ ইফ এলস স্টেটমেন্ট (If -- Else)	১৪
৭.৮ এলস ইফ স্টেটমেন্ট (Else -- If)	১৪
৭.৯ সুয়িচ স্টেটমেন্ট (Switch)	১৪
৮. লুপের প্রকার ভেদ, ফরলুপ, হোয়াইল লুপ, ডু-হোয়াইল লুপ, ফর ইচ লুপ সম্পর্কে আলোচনা	১৪
৮.১ ভূমিকা	১৪
৮.২ লুপের প্রকার ভেদ	১৪
৮.৩ ফরলুপ	১৪
৮.৪ ফরলুপ (For loop)	১৪
৮.৫ হোয়াইল লুপ (while loop)	১৪
৮.৬ ডু-হোয়াইল লুপ (do while loop)	১৪
৮.৭ ফর ইচ লুপ (For each loop)	১৪
৯. ব্রেক, কনটিনিউটি, স্ট্রিং এ্যান্ড কনক্যাটেনেশন	১৪
৯.১ ভূমিকা :	১৪
৯.২ ব্রেক (Break) কি	১৪
৯.৩ ব্রেক এর সিনট্যাক্স (Syntax)	১৪
৯.৪ ব্রেক এর উদাহরণ	১৪
৯.৫ কনটিনিউ (continue)	১৪
৯.৬ Continue এর syntax	১৪
৯.৭ কনটিনিউ স্টেটমেন্ট এর উদাহরণ (Statement)	১৪
৯.৮ স্ট্রিং (String)	১৪
৯.৯ কনক্যাটেনেশন (Concatenation)	১৪
৯.১০ অধ্যায় সার সংক্ষেপ	১৪
৯.১১ প্রশ্ন	১৪
১০. অ্যারে (Array)	১৪
১০.১ ভূমিকা :	১৪
১০.২ অ্যারে (Array) কি?	১৪
১০.৩ অ্যারে এর স্ট্রাকচার (Structure)	১৪
১০.৪ অ্যারে এর প্রকারভেদ (Type of Array)	১৪
১০.৫ নিউমেরিক (Number) অ্যারে	১৪
১০.৬ অ্যাসোসিয়েটিভ (Associative) অ্যারে	১৪

১০.৭ মাল্টি-ডাইমেনশনাল অ্যারে (Multidimensional Arrays) -----	১৪
১০.৮ অ্যারে ফাংশন (Php Array Function) -----	১৪
১০.৯ সংক্ষেপে অধ্যায় -----	১৪
১০.১০ প্রশ্ন -----	১৪
১১. ফাংশন (Function) -----	১৪
১১.১ ভূমিকা : -----	১৪
১১.২ ফাংশন (Function) কি? -----	১৪
১১.৩ ফাংশন লিখার নিয়ম (Syntax of Function) -----	১৪
১১.৪ কলিং ফাংশন (Calling Function) -----	১৪
১১.৫ ফাংশন এর উদাহরণ -----	১৪
১১.৬ ফাংশন কেন ব্যবহার করা হয়? -----	১৪
১১.৭ এক নজরে অধ্যায় -----	১৪
১১.৮ প্রশ্ন -----	১৪
১২. মেথড (Methods) -----	১৪
১২.১ ভূমিকা -----	১৪
১২.২ মেথড কি (Method) -----	১৪
১২.৩ মেথড সিনটাক্স -----	১৪
১২.৪ মেথড তৈরী -----	১৪
১২.৫ মেথড কল -----	১৪
১২.৬ মেথড প্যারামিটার -----	১৪
১২.৭ মেথড প্যারামিটার পাঠানো -----	১৪
১২.৮ মেথড রিটার্ন টাইপ -----	১৪
১২.৯ অধ্যায় সারাংশ -----	১৪
১২.১০ অনুশীলন প্রোগ্রাম -----	১৪
১২.১১ প্রশ্নমালা -----	১৪
১৩. মডিফায়ার এবং কী-ওয়ার্ড (Modifiers) -----	১৪
১৩.১ ভূমিকা -----	১৪
১৩.২ মডিফায়ার কি? -----	১৪
১৩.৩ মডিফায়ার লিখার নিয়ম -----	১৪
১৩.৪ পাবলিক (public) -----	১৪
১৩.৫ প্রাইভেট (Private) -----	১৪
১৩.৬ প্রোটেক্টেড (Protected) -----	১৪
১৩.৭ অ্যাবসট্রাক্ট এবং ফাইনাল (Abstract Final) -----	১৪
১৩.৮ এক নজরে অধ্যায় -----	১৪
১৩.৯ প্রশ্ন -----	১৪

১৪. কনস্ট্রাকটর এবং ডেস্ট্রাকটর (Constructor & Destructor)	১৪
১৪.১ ভূমিকা	১৪
১৪.২ কনস্ট্রাকটর	১৪
১৪.৩ কনস্ট্রাকটর লিখার নিয়ম	১৪
১৪.৪ কনস্ট্রাকটর এর প্রকারভেদ (Type of Constructor)	১৪
১৪.৫ ডেস্ট্রাকটর (Destructor)	১৪
১৪.৬ একনজরে অধ্যায়	১৪
১৪.৭ প্রশ্ন	১৪
১৫. ইনহেরিটেন্স (Inheritance)	১৪
১৫.১ ভূমিকা	১৪
১৫.২ ইনহেরিটেন্স কি?	১৪
১৫.৩ ইনহেরিটেন্স কিভাবে কাজ করে?	১৪
১৫.৪ ইনহেরিটেন্স এর সুবিধা	১৪
১৫.৫ ইনহেরিটেন্স এর অসুবিধা গুলো	১৪
১৫.৬ সিনট্যাক্স ইনহেরিটেন্স (Inheritance Syntax)	১৪
১৫.৭ দিস (\$this) ওয়ার্ড	১৪
১৫.৮ ইনহেরিটেন্স এবং মডিফায়ার	১৪
১৫.৯ একনজরে অধ্যায়	১৪
১৫.১০ প্রশ্ন	১৪
১৬. পলিমরফিজম (Polymorphism)	১৪
১৬.১ ভূমিকা	১৪
১৬.২ পলিমরফিজম (Polymorphism) কি?	১৪
১৬.৩ পলিমরফিজম যেভাবে কাজ করে	১৪
১৬.৪ পিএইচপি পলিমরফিজমের প্রোগ্রাম সমূহ	১৪
১৬.৫ একনজরে অধ্যায়	১৪
১৬.৬ প্রশ্ন	১৪
১৭. ইনক্যাপসুলেশন (Encapsulation)	১৪
১৭.১ ভূমিকা	১৪
১৭.২ ইনক্যাপসুলেশন কি?	১৪
১৭.৩ ইনক্যাপসুলেশন এর সুবিধা	১৪
১৭.৪ ইনক্যাপসুলেশন কিভাবে কাজ করে?	১৪
১৭.৫ ইনক্যাপসুলেশন প্রোগ্রামিং	১৪
১৭.৬ একনজরে অধ্যায়	১৪
১৭.৭ প্রশ্ন	১৪
১৮. অ্যাবস্ট্রাক্ট (Abstract)	১৪
১৮.১ ভূমিকা	১৪
১৮.২ অ্যাবস্ট্রাক্ট (Abstract) কি?	১৪
১৮.৩ অ্যাবস্ট্রাক্ট ক্লাস যেভাবে কাজ করে	১৪

১৮.৪ অ্যাবস্ট্র্যাক প্রোগ্রামিং সমূহ -----	১৪
১৮.৫ অ্যাবস্ট্র্যাক ক্লাসের সুবিধা -----	১৪
১৮.৬ এক নজরে অধ্যায় -----	১৪
১৮.৭ প্রশ্ন -----	১৪
১৯. ইন্টারফেইস (Interface) -----	১৪
১৯.১ ভূমিকা -----	১৪
১৯.২ ইন্টারফেইস (Interface) কি? -----	১৪
১৯.৩ ইন্টারফেইস যেভাবে কাজ করে -----	১৪
১৯.৪ ইন্টারফেইস এর সুবিধা গুলো -----	১৪
১৯.৫ ইন্টারফেইস প্রোগ্রামিং -----	১৪
১৯.৬ এক নজরে অধ্যায় -----	১৪
১৯.৭ প্রশ্ন -----	১৪
২০. পিএইচপি ফর্ম (ইনপুট এবং আউটপুট) -----	১৪
২০.১ এখানে আমরা দেখবো অবজেক্ট অরিয়েন্টেড পদ্ধতিতে কিভাবে পিএইচপি ইনপুট নেওয়া যায় এবং আউটপুট দেখানো যায় -----	১৪
২০.২ স্টুডেন্ট রেজিস্ট্রেশন -----	১৪
২১. পিএইচপি মাইএসকিউএল প্রজেক্ট (input/update/delete) -----	১৪
২১.১ ডেটাবেইজ ভ্যালু ইনসার্ট (insert) অবজেক্ট অরিয়েন্টে পিএইচপি ব্যবহার করে কিভাবে ডেটাবেইজে ভ্যালু সংরক্ষণ (Store) করা যায় এখন আমরা সেটা দেখবো -----	১৪
২১.২ ডিসপ্লে ডেটাবেইজ ভ্যালু -----	১৪
২১.৩ আপডেট এবং ডিলিট ইনফরমেশন (Edit/Delete data) -----	১৪
২২. টেক্সবক্স ভ্যালু এবং ডেটাবেইজ -----	১৪
২২.১ একটি টেক্সবক্স থেকে ভ্যালু নিয়ে কিভাবে ডেটাবেইজে রাখতে হবে সেটা দেখবো -----	১৪
২৩. রেজিস্ট্রেশন সিস্টেম -----	১৪
২৩.১ এখানে আমরা আপনাকে খুব সহজ একটি রেজিস্ট্রেশন পদ্ধতি দেখাবো যেটি দেখে আপনি খুব সহজে অন্য যে কোন ধরনের রেজিস্ট্রেশন (Registration) এবং ডেটা ইনসার্ট আপডেট এবং ডিলিট করতে পারবেন --	১৪

অধ্যায়- ১ অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং

এই অধ্যায়ের আলোচ্য বিষয় সমূহ

১. অবজেক্ট অরিয়েন্টেড পিএইচপি এর প্রাথমিক আলোচনা -----
- ১.১ ভূমিকা -----
- ১.২ অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং এ সূচনা -----
- ১.৩ অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং কি? -----
- ১.৪ অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং ল্যাংগুয়েজ সমূহ -----
- ১.৫ অবজেক্ট অরিয়েন্টেড এবং প্রোসিডিউর ল্যাংগুয়েজ এর মধ্যে পার্থক্য -----
- ১.৬ অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং এর সুবিধা -----
- ১.৭ ক্যারিয়ার হিসেবে অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং -----
- ১.৮ অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং এর জটিলতা -----
- ১.৯ বহুল ব্যবহৃত অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং শব্দ সমূহ -----
- ১.১০ এক নজরে অধ্যায় সারাংশ -----
- ১.১১ প্রশ্ন সমূহ -----

১.১ ভূমিকা :

কম্পিউটার প্রোগ্রামিং হচ্ছে এমন এক ধরনের প্রোগ্রাম যেখানে একের পর এক (Sequence) কত গুলো নির্দেশনা (instruction) দেওয়া থাকে। একেকটি নির্দেশনা (নির্দেশ) এর উদ্দেশ্য একেক ধরনের। অর্থাৎ আমাদের বাস্তব জীবনের বিভিন্ন সমস্যার সমাধান করার জন্য কম্পিউটারে এই সব নির্দেশনা গুলো দেওয়া হয় আর এই প্রক্রিয়াটিকে বলা হয় প্রোগ্রামিং। আর ক গুলো Instruction (নির্দেশ) কে আমরা প্রোগ্রামিং বলে থাকি যে গুলো কম্পিউটার এ আমরা সম্পাদন (run) করে থাকি। বর্তমানে অনেক ধরনের প্রোগ্রামিং ল্যাংগুয়েজ প্রচলিত রয়েছে যেমনঃ সি, সি++, জাভা, পিএইচপি ইত্যাদি। এই বইয়ে আমরা আলোচনা করব অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং পিএইচপি।

১.২ অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং এর সূচনা

১৯৬০ সালের দিকে প্রথম অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং এর ধারণা আসে। সিমুলা (Simula) হচ্ছে প্রথম দিকের অবজেক্ট অরিয়েন্টেড ল্যাংগুয়েজ। সিমুলা ল্যাংগুয়েজটি ডিজাইন করেছেন ওলে জোহান এবং ক্লিসটেন নাইগার্ড।

১.৩ অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং কি?

আমাদের চারদিকে আমরা যা দেখি যেমন গাড়ী, বাড়ী, গাছ, বুক, টেবিল, পেন ইত্যাদি প্রোগ্রামিং এ এগুলোকে অবজেক্ট বলে অর্থাৎ অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং অবজেক্টকে উপস্থাপন করে। অবজেক্ট এর ডেটাফিল্ড এবং অ্যাক্টিভিউট থাকে। সফটওয়্যার/প্রোগ্রামিং এ অবজেক্টকে প্রায় ব্যবহার করা হয় বাস্তব জীবনের (Real Life) কোন কিছুকে মডেল হিসেবে উপস্থাপন করার জন্য। OOP হচ্ছে এমন এক ধরনের ল্যাংগুয়েজ সেটি শুধুমাত্র ডেটা নিয়ে কাজ করে না এটি বিভিন্ন ধরনের অপারেশন করার জন্য ফাংশন ব্যবহার করে ডেটা স্ট্রাকচার (Data Structure) এর উপর। আর এখানে ডেটা স্ট্রাকচার হচ্ছে অবজেক্ট যেখানে ডেটা এবং ফাংশন দুটোই থাকে। অর্থাৎ অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং কত গুলো অবজেক্ট নিয়ে কাজ করে। এই বইটি পড়ার পূর্বে অবশ্যই বুকবিডি সিরিজের সি এবং পিএইচপি বইটি পড়তে হবে।

১.৪ অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং ল্যাংগুয়েজ সমূহ

জনপ্রিয় অবজেক্ট অরিয়েন্টেড ল্যাংগুয়েজ সমূহঃ

১. পিএইচপি (PHP)
২. সি++(C++)
৩. জাভা (Java)
৪. ডেলপি (Delphi)
৫. পাইথন (Python)
৬. ভিজুয়াল বেসিক (Visual Basic)
৭. সি সার্প (C#)

১.৫ অবজেক্ট অরিয়েন্টেড এবং প্রোসিডিউর ল্যাংগুয়েজ এর মধ্য পার্থক্য ।

১.

- > প্রোসিডিউর অরিয়েন্টেড প্রোগ্রামিং এর ছোট ছোট ভাগ গুলোকে ফাংশন বলে ।
- > অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং এর এক একটি ভাগকে অবজেক্ট বলে ।

২.

- > প্রোসিডিউর প্রোগ্রামিং এ শেড গুলো ছোট ছোট পার্ট পার্টে সাজানো থাকে যেটিকে আমরা প্রোসিডিউর (Procedure) বলে থাকি ।
- > অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং এ ডেট এবং এর সাথে সংশ্লিষ্ট ফাংশন গুলো এক সাথে সংযুক্ত থাকে । আর এটিই হচ্ছে অবজেক্ট ।

৩. প্রোসিডিউর প্রোগ্রামিং এ কতগুলো নির্দেশ (Instruction) থাকবে যে গুলো কম্পিউটারকে একের পর এক কিছু করতে বলবে । অর্থাৎ প্রথম লাইন কোড এর পর দ্বিতীয় লাইন তারপর তৃতীয় লাইন এভাবে চলতে থাকবে শেষ লাইন পর্যন্ত ।

- > অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং, প্রোগ্রামিং এর কাজ গুলোকে ভেঙ্গে অবজেক্ট এ নিয়ে আসে ।

৪. প্রোসিডিউর ল্যাংগুয়েজ টপ ডাউন (Top Down) অ্যাপ্রোচ অনুসারে কাজ করে ।

- > অবজেক্ট অরিয়েন্টেড বটম আপ অ্যাপ্রোচ অনুসারে কাজ করে ।

৫. প্রোসিডিউর ল্যাংগুয়েজে ডেটা অথবা ফাংশনকে ব্যবহার করতে কোন বিধি নিষেধ (Access Specifier) নেই ।

- > অবজেক্ট অরিয়েন্টেড ল্যাংগুয়েজে ডেটা মেথড এবং অন্যান্য কিছু ব্যবহার করার ক্ষেত্রে বিভিন্ন ধরনের প্রাইভেসি (বিধি নিষেধ) আছে ।

১.৬ অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং এর সুবিধা :

১. অবজেক্ট গুলোকে খুব সহজে মেনটেইন করা যায় ।

২. এটি সহজ হওয়ার কারণে সফটওয়্যার এর অবজেক্ট মডেল এর সাথে বাস্তব জীবনের অবজেক্ট এর সাথে মিল আছে ।

৩. অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং এর কোড গুলো পুনরায় ব্যবহার করা যায় । অর্থাৎ একই অবজেক্ট বিভিন্ন জায়গায় ব্যবহার করা যায় ।

৪. অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং ক্লাস, ইনক্যাপসুলেশন, ইনহেরিটেন্স পলিমরপিজম ইত্যাদি সার্পোর্ট করে ।

৫. এটি খুবই Secured

৬. সমস্যার উপর নির্ভর করে মূল প্রোগ্রামকে ভেঙ্গে ভেঙ্গে অবজেক্ট এ রূপান্তরিত করে ।

৭. ডেটা হাইড করা সম্ভব ।

৮. প্রোসিডিউর ল্যাংগুয়েজ কাজ করে ফাংশন নিয়ে । আর অবজেক্ট অরিয়েন্টেড ল্যাংগুয়েজ কাজ করে অবজেক্ট এবং ক্লাস নিয়ে ।

৯. প্রোসিডিউর ল্যাংগুয়েজের নিরাপত্তা (Security) অনেক কম । আর অবজেক্ট অরিয়েন্টেড ল্যাংগুয়েজে সিকিউরিটি অনেক বেশি ।

১০. এটিকে মডিফাই করা অনেক সহজ

১১. অনেক বড় বাস্তব জীবনের সমস্যাকে এটির মাধ্যমে খুব সহজে সমাধান করা সম্ভব ।

১২. খুব সহজে একটি কোডের (প্রোগ্রামের) এর সাথে নতুন (ফিচার) সংযোগ করা যায় ।

১১. এটি ব্যবহার করলে প্রোগ্রাম এর জটিলতা অনেক কমে যায়।

১.৭ ক্যারিয়ার হিসেবে অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং

বর্তমানে ক্যারিয়ার হিসেবে আপার হাই (High) লেভেল ল্যাংগুয়েজে বেচে নেওয়া উত্তম। বর্তমানে বেশির ভাগ হাই লেভেল (High Level) ল্যাংগুয়েজ অবজেক্ট অরিয়েন্টেড ভিত্তিক। এক সমীক্ষায় দেখা গেছে বেশির ভাগ চাকুরির ক্ষেত্রে অবজেক্ট অরিয়েন্টেড প্রোগ্রাম জানা অত্যাবশ্যক। বর্তমানে সফটওয়্যার এর সিকিউরিটি অত্যন্ত গুরুত্বপূর্ণ একটি বিষয় তাই অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং দিয়ে তৈরিকৃত সফটওয়্যার বেশি নিরাপত্তা দেওয়া সম্ভব। বর্তমান বাজারে চাকুরির ক্ষেত্রে যে ল্যাংগুয়েজ গুলো চাওয়া হয় যেমন : পিএইচপি, জাভা, পাইথন, রুবি, সি #, ডটনেট, সি++, ইত্যাদি জনপ্রিয় ল্যাংগুয়েজ গুলো সবই অবজেক্ট অরিয়েন্টেড ভিত্তিক।

১.৮ অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং এর জটিলতা

১. টেকনোলজি পরিবর্তনের সাথে আপনাকে পরিবর্তিত বিষয় সমূহ জেনে নিতে হবে।
২. ইনহেরিটেন্স এর ক্ষেত্রে সাব ক্লাস সম্পূর্ণ ভাবে সুপার ক্লাস এর উপর নির্ভরশীল।
৩. এটি শিখতে হলে বেসিক প্রোগ্রামিং লজিক সম্পর্কে খুব ভালো ধারণা থাকতে হবে।
৪. কিছু কিছু ক্ষেত্রে এটি রান হতে কিছু বেশি সময় নেয়।
৫. অবজেক্ট অরিয়েন্টেড উপায়ে সিস্টেম ডিজাইন এর ক্ষেত্রে কিছু জটিলতা দেখা যায়।
৬. প্রোগ্রাম এ কোন ধরনের পরিবর্তন আনলে ক্লাস এবং অবজেক্ট দুটোই পরিবর্তন হয়ে যায় (অর্থাৎ কোন সমস্যার পরিবর্তন হলে)

১.৯ বহুল ব্যবহৃত অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং এর শব্দ সমূহ

অবজেক্ট (Object) : আমাদের চারদিকে যা দেখি তাই হচ্ছে অবজেক্ট। যেমনঃ বই, কলম, কার (গাড়ি), মানুষ, বালিকা ইত্যাদি। এটি কোন তথ্যকে ধারণ অথবা উপস্থাপন করে।

ক্লাস (Class) : অবজেক্টকে ক্লাসে ডিফাইন করা হয়। অর্থাৎ অবজেক্ট সম্পর্কে বিস্তারিত ক্লাসে বলা থাকে। অবজেক্ট এর বর্ণনা (বিস্তারিত) ক্লাসে থাকে।

মেথড (Method) : মেথড হচ্ছে এক ধরনের ফাংশন। একই ধরনের কাজ বার বার করার জন্য মেথড ব্যবহার করা হয়। মেথড অনেকটা প্রসিডিউর (Subroutine) এর মত কাজ করে। মেথড শুধু রান টাইমে অ্যাকসেস করা যায়।

প্যারামিটার (Parameter) : প্যারামিটার হচ্ছে বিশেষ ধরনের ভেরিয়েবল। অর্থাৎ প্রোগ্রাম এর ভিতর এক স্থান হতে অন্য স্থানে ভ্যালু (Value) পাঠানোর জন্য প্যারামিটার ব্যবহার করা হয়।

কনস্ট্রাক্টর (Constructor) : এটি হচ্ছে বিশেষ ধরনের ফাংশন। এটিকে কল করতে হয় না। প্রোগ্রাম রান করলে এটি অটোমেটিক্যালি রান করে।

মডিফায়ার (Modifier) : এটি ব্যবহার করা হয় ক্লাস, মেথড ফাংশন ভ্যারিয়েবল ইত্যাদির ব্যবহার এর উপর অ্যাকসেস কন্ট্রল করার জন্য। এটি হচ্ছে কিছু কী ওয়ার্ড। এ ধরনের মডিফায়ার গুলো যেমন পাবলিক (Public) প্রাইভেট (Private) প্রোটেক্টেড (Protected)।

১.১০ এক নজরে অধ্যায় সারাংশ

অবজেক্ট অরিয়েন্টেড ল্যাংগুয়েজ গুলোর চাহিদা ক্রমে বেড়ে চলছে। যদি ও ১৯৬০ সালের দিকে সিমুলা ল্যাংগুয়েজ এর মাধ্যমে এর যাত্রা শুরু হয়। মূলত অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং বাস্তব জীবনের সাথে মিল রেখে তৈরী করা। বালক, চেয়ার, টেবিল ইত্যাদি। বর্তমানে জনপ্রিয় অবজেক্ট অরিয়েন্টেড ল্যাংগুয়েজ গুলো হল পিএইচপি, জাভা, সি#, সি++, পাইথন ইত্যাদি।

> প্রোসিডিউর প্রোগ্রামিং এ কত গুলো ইনস্ট্রাকশন দেওয়া থাকবে যে গুলো কম্পিউটারে একের পর এক রান করবে। অন্য দিকে অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং এক একটি ছোট ভাগকে (প্রোগ্রাম এর) অবজেক্ট বলে। অবজেক্ট অরিয়েন্টেড সুবিধা যেমন একই কোড পুনরায় ব্যবহার করা যায়। বাস্তব জীবনের অবজেক্ট এর সাথে এর মিল আছে। অবজেক্ট গুলো খুব সহজে মেনটেইন করা যায়। যদি ও নতুনদের জন্য এটি শিখা কিছুটা জটিল। বর্তমানে বাজারে এর রয়েছে ব্যাপক চাহিদা তাই ক্যারিয়ার হিসেবে এটি একটি সর্বোত্তম ল্যাংগুয়েজ।

> অবজেক্ট কত গুলো প্রোপার্টিজ এবং মেথড কে ধারণ করে। অবজেক্ট এর Propertise এর মাধ্যমে একটি অবজেক্ট সম্পর্কে জানতে পারি। আর ক্লাস হচ্ছে একটি প্রোগ্রাম যেখানে কতগুলো নির্দেশ (Instruction) দেওয়া থাকে অবজেক্ট সম্পর্কে।

১.১১. প্রশ্ন

১. অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং কি?
২. অবজেক্ট কাকে বলে?
৩. ক্লাস কি?
৪. অবজেক্ট এবং ক্লাসের মধ্য পার্থক্য লিখ।
৫. অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং এর বেসিক কনসেপ্ট গুলো লিখ।
৬. অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং এর সুবিধা সমূহ কি কি?
৭. অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং এর বৈশিষ্ট্য গুলো লিখ।
৮. অবজেক্ট অরিয়েন্টেড এবং প্রোসিডিউর ল্যাংগুয়েজ এর মধ্য পার্থক্য লিখ।
৯. প্রথম অবজেক্ট অরিয়েন্টেড ল্যাংগুয়েজটির নাম কি?
১০. ১০টি অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং এর নাম লিখ।
১১. অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং ক্যারিয়ার এর জন্য কেন উত্তম।

Quick & Easy Way

Your Professional Trainer...

নিজে নিজে বিজনেস অ্যাপ্লিকেশন তৈরি করুন

অ্যাডভান্স মাইক্রোসফট এক্সেল-২০১৩

এক্সেল বেসিক
ডেটা এন্ট্রি ফর্ম তৈরি
এক্সেল বোম্বাস কাস্টোমাইজেশন
ফরম্যাটিং টিপস
ফর্মুলা এবং ফাংশন
চার্ট এবং গ্যাঞ্চার
ম্যাক্রো এবং ডেটাবেজ ম্যানেজমেন্ট

রচনা ও সম্পাদনা :
বুকবিডি

১০ টি বাস্তবধর্মী প্রোজেক্ট

বর্তমান টেকনোলজির যুগে আমরা আমাদের দৈনন্দিন কাজ যেমন বাজিগত ডকুমেন্ট ব্যবসায়িক হিসাব, পরিসংখ্যান এবং লজিক্যাল সবই এক্সেল ব্যবহার করে তৈরি করা সম্ভব তাই খুব সহজে নিজে নিজে এক্সেল শিখার জন্য বুকবিডি সিরিজের এই বইটি।

With DVD

Download

Latest Version, 2013, 2010, 2007 www.bookbd.info

More Than Just a Book

অধ্যায়-২ অবজেক্ট অরিয়েটেড পিএইচপি

এই অধ্যায়ের আলোচ্য বিষয় সমূহ

২. পিএইচপি ও অবজেক্ট অরিয়েটেড সম্পর্কে আলোচনা -----
- ২.১ ভূমিকা -----
- ২.২ পিএইচপি কি? -----
- ২.৩ পিএইচপি ল্যাংগুয়েজ কিভাবে কাজ করে? -----
- ২.৪ ক্যারিয়ার হিসেবে অবজেক্ট অরিয়েটেড পিএইচপি -----
- ২.৫ অবজেক্ট অরিয়েটেড পিএইচপি এর সুবিধা -----
- ২.৬ ল্যাম্প -----
- ২.৭ অধ্যায় সারাংশ -----
- ২.৮ প্রশ্ন -----

২.১ ভূমিকা :

পিএইচপি হচ্ছে একটি হাইলেভেল ল্যাংগুয়েজ কিন্তু অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং মডেল এর উপর ভিত্তি করে এটি তৈরী। পিএইচপি কে বলা হয় শক্তিশালী সার্ভার সাইড Scripting Language। সফটওয়্যার ডেভেলপমেন্ট ইন্ডাস্ট্রিতে রয়েছে এর ব্যাপক চাহিদা। এটি একটি ওপেন সোর্স ল্যাংগুয়েজ।

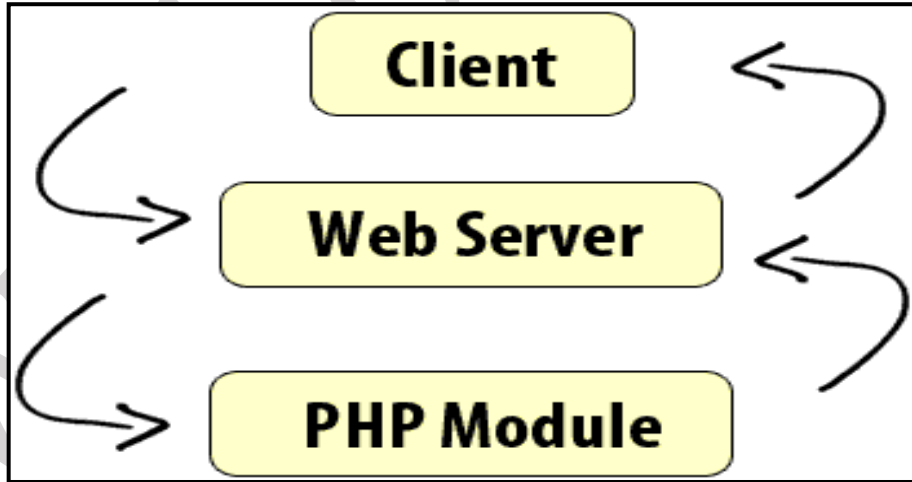
২.২ পিএইচপি কি?

পিএইচপি হচ্ছে একটি স্ক্রিপ্টিং ল্যাংগুয়েজ এটা রান (Execute) করা হয় সার্ভারে। PHP হচ্ছে হাইপারটেক্স প্রিপ্রেসেসর (Hypertext Preprocessor) প্রোগ্রামিং ল্যাংগুয়েজ যেটি দিয়ে ডাইনামিক ওয়েব সাইট অর্থাৎ ওয়েব বেসড অ্যাপ্লিকেশন তৈরী করা হয় এবং এটি ডেটাবেইজ এর সাথে ও কাজ করে। ওয়েব বেসড সফটওয়্যার তৈরীর জন্য পিএইচপি হচ্ছে একটি শক্তিশালী (Powerful) টুলস।

২.৩ পিএইচপি ল্যাংগুয়েজ কিভাবে কাজ করে

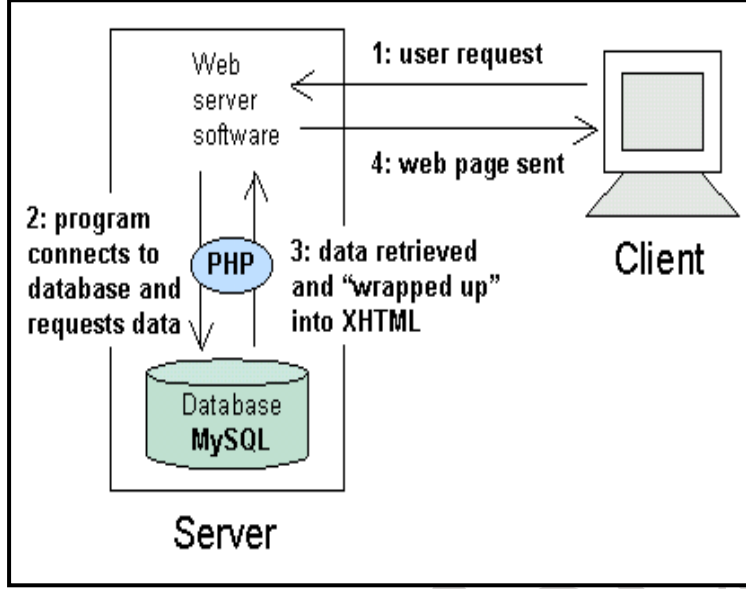
> যেহেতু পিএইচপি একটি সার্ভার সাইড (Server Side) স্ক্রিপ্টিং ল্যাংগুয়েজ সে জন্য এটি ওয়েব সার্ভার ছাড়া রান (Execute) করবেনা। এই বইটি ভালো ভাবে বুঝতে হলে বুকবিডি সিরিজের পিএইচপি এবং মাইএসকিউ এল বইটি অবশ্যই পড়বেন। তাই PHP জন্য আপনার একটা সার্ভার কম্পিউটার লাগবে। চাইলে আপনার কম্পিউটারে অ্যাপাচি (Apache) নামক সফটওয়্যার ইনস্টল করে সার্ভার বানাতে পারেন।

> ভিজিটর যখন একটি পেইজ ওপেন (Request) করে তখন সেটি (Request টি) সার্ভারে যায় সার্ভার পিএইচপি কোডকে প্রসেস করে সে অনুযায়ী আউটপুট হিসেবে ব্রাউজারে এইচটিএমএল দ্বারা লিখিত কনটেন্ট সমূহ প্রদর্শিত করে। এইচটিএমএল দ্বারা লিখা কোন পেইজ (ওয়েব পেইজ) এর কোড ইউজার তার ব্রাউজারে দেখতে পারে। কারণ এ কোড গুলো ব্রাউজারে রান (Interpret) করে। কিন্তু পিএইচপি দিয়ে লিখা কোন পেইজের কোড ব্রাউজারে শো (Show) করে না। কারণ এটি রান করে ওয়েব সার্ভারে (Apache অথবা IIS)।



চিত্রঃ ২.৩.১ পিএইচপি কাজ করার একটি প্রক্রিয়া

নিম্নে পিএইচপি কিভাবে ক্লায়েন্ট ওয়েবসার্ভার এবং ডেটাবেইজ (Database) নিয়ে কাজ করে তা দেখানো হলো।



(চিত্রঃ ২.৩.২ ক্লায়েন্ট সার্ভার রিলেশন)

উপরের চিত্রে দেখা যাচ্ছে ক্লায়েন্ট (User) কম্পিউটার তার নিজের কম্পিউটার থেকে ওয়েব সার্ভার থেকে একটি অনুরোধ (user request) করল এবং Request টি ওয়েব সার্ভারে চলে গেল। ওয়েব সার্ভার রিকোয়েস্ট PHP কে প্রসেস করতে দিল। এখানে পিএইচপি তার প্রয়োজনে ডেটাবেইজ এর সাথে কানেক্ট হতে পারে (2 Program Connects to database and request data)। পিএইচপি ডেটাকে প্রসেস করে হতে পারে পিএইচপি এই ডেটা তথ্য ডেটাবেইজ থেকে নিয়ে এসেছিল এবং সেটাকে ওয়েব সার্ভারকে দিল (3: Data retrieved and wrapped up into XHTML/HTML) ওয়েব সার্ভার ডেটা টা পেয়ে ক্লায়েন্টকে ওয়েব পেইজ এর মাধ্যমে প্রদর্শিত করল। নিম্নে পিএইচপি Interpret করার পদ্ধতি নিয়ে আরেকটি চিত্র (সংগৃহীত) দেওয়া হল।

উপরের চিত্রে লক্ষ্য করুন ইউজার যখন কোন ব্রাউজারের মাধ্যমে সার্ভারকে কোন পেইজ এর জন্য রিকোয়েস্ট করে আর যদি ঐ ফাইলটির Extension (.php) হয়। অর্থাৎ সেটি যদি কোন পিএইচপি ফাইল হয়ে থাকে। তখন ঐ Request টি পিএইচপি Interpreter এর কাছে চলে যায়। যেটি উপরের চিত্রে লক্ষ্য করেছেন। পিএইচপি Interpreter তখন ঐ ফাইল সিসটেম এর সাথে যোগাযোগ (Communicate) করে ঐ ফাইলটির জন্য (সকল PHP ফাইল ফাইল সিসটেম এ থাকে)। ওয়েব পেইজ ডিজাইন এবং ডেভেলপমেন্ট শিখার জন্য বুকবিডির এইচটিএমএল বইটি পড়তে পারেন। এ ছাড়া ও প্রয়োজনে মেইল সার্ভার, ডেটাবেজ এর সাথে যোগাযোগ করে। পরবর্তীতে PHP পেইজটি ওয়েব সার্ভারে পাঠানো হয় এবং সর্বশেষ ওয়েব সার্ভার পেইজটি ব্রাউজারে পাঠিয়ে দেয়।

২.৪ ক্যারিয়ার হিসেবে অবজেক্ট অরিয়েন্টেড পিএইচপি

বর্তমান বিশ্বে ওয়েব ডেভেলপমেন্ট, ডাইনামিক ওয়েব সাইট, ওয়েব বেইসড সফটওয়্যার অ্যাপ্লিকেশন তৈরী জন্য অত্যন্ত জনপ্রিয় একটি ল্যাংগুয়েজ হচ্ছে পিএইচপি। সারা বিশ্বে রয়েছে এর ব্যাপক জব চাহিদা এবং অনেক বেশী আকর্ষণীয় বেতন। অনলাইনে তথা আউটসোর্সিং কাজের ক্ষেত্রে ও রয়েছে এর ব্যাপক চাহিদা অনলাইনে আয় করতে চাইলে বুকবিডির ইন্টারনেটে আত্মকর্মসংস্থান বইটি পড়তে পারেন। পিএইচপি প্রোগ্রামার তাদের অভিজ্ঞতার মাধ্যমে খুব সহজে একজন সফটওয়্যার ইঞ্জিনিয়ার হতে পারেন। আবার কিছু কিছু ফার্ম শুরু থেকে পিএইচপি প্রোগ্রামারদেরকে Software Engineer হিসেবে নিয়োগ দিয়ে থাকে। পিএইচপিকে ক্যারিয়ার হিসেবে পছন্দ করার ১০টি কারণ নিম্নে দেওয়া হল।

১. বিশ্বেও এক হাজারের ও বেশি Communities থেকে পিএইচপি সংক্রান্ত যে কোন ধরনের সাহায্য পাওয়া যায়। বিভিন্ন ধরনের প্রশ্ন করলে অতি দ্রুত উত্তর পাওয়া যায়।
২. পিএইচপি যেহেতু একটি ওপেন সোর্স ল্যাংগুয়েজ সুতরাং এটি ক্রয় করতে হয় না। শুধুমাত্র ডেভেলপমেন্ট খরচ আছে।
৩. সারা বিশ্বে যখন ওয়েব সাইট এর সিকিউরিটি নিয়ে উদ্দিগ্ন সেখানে পিএইচপি ল্যাংগুয়েজ দিচ্ছে সন্তোষজনক সিকিউরিটি।
৪. ফ্রিল্যান্সিং আউটসোর্সিং এ রয়েছে এর ব্যাপক চাহিদা।
৫. বর্তমানে পিএইচপি এর অনেক ফ্রেমওয়ার্ক রয়েছে। যে গুলোর মাধ্যমে ওয়েব অ্যাপ্লিকেশন (Software) বানানো অনেক সহজ।
৬. পিএইচপি হচ্ছে খুব শক্তিশালী একটি টুলস এবং Web 2.0 ফ্রেন্ডলি।
৭. সারা বিশ্ব এখন সিএমএস (CMS) নির্ভর হচ্ছে এবং বেশির ভাগ CMS গুলো পিএইচপি দিয়ে তৈরী যেমন জুমলা, ড্রুপাল, ওয়াডপ্রেস ইত্যাদি।
৮. একটি সফটওয়্যার একবার ডেভেলপ করার পর বিভিন্ন সময় পরিবর্তন করতে হয়। আর পিএইচপি দিয়ে তৈরী করা সফটওয়্যার গুলো খুব সহজে পরিবর্তন করা যায়।
৯. বেশির ভাগ কোম্পানী পিএইচপি নিয়ে কাজ করছে। আর এজন্য PHP ডেভেলপারদের চাহিদা দিন দিন বেড়ে চলছে।
১০. পিএইচপি কোড অন্যান্য ল্যাংগুয়েজের চেয়ে অনেক সহজ। অনেকটা বলা যায় Painless প্রোগ্রামিং।

২.৫ অবজেক্ট অরিয়েন্টেড পিএইচপি এর সুবিধা

১. পিএইচপি একটি এ্যাকসেসিবল (Accessible) ল্যাংগুয়েজ।
২. এটি একটি ওপেন সোর্স ল্যাংগুয়েজ অর্থাৎ বিনামূল্যে পাওয়া যায়।
৩. এটির মাধ্যমে অনেক দ্রুত সফটওয়্যার এ্যাপ্লিকেশন তৈরী করা সম্ভব।
৪. এটিকে প্রায় সব ধরনের অপারেটিং সিস্টেমে এ রান (Execute) করা যায়।
৫. পিএইচপি ল্যাংগুয়েজের সিনট্যাক্স গুলো শিখা অনেক সহজ।
৬. অনেক সহজে পিএইচপি এবং মাইএসকিউএল এর সাথে ইন্টারফেসিং করা যায়।
৭. অনেক হোস্টিং কোম্পানি রয়েছে যারা অনেক কম মূল্যে হোস্টিং দিচ্ছে।
৮. এটি আপনার সিস্টেম এর গতি কমাবেনা।
৯. এটি খুব দ্রুত লোড হয়।
১০. পিএইচপি ফাইলে এইচটিএমএল অন্তর্ভুক্ত করা যায়।
১১. টেক্স এডিটর (text editor) এবং নোট প্যাড (Notepad) এ পিএইচপি কোড লিখা এবং এডিট করা যায়।
১২. বর্তমানে টেকনোলজি গুলোর মধ্য স্ট্যান্ডার্ড স্ক্রিপ্টিং ল্যাংগুয়েজ।
১৩. ওয়েব ডেভেলপমেন্ট এর ফ্রন্টয়েন্ড এবং ব্যাকয়েন্ড উভয় ক্ষেত্রে পিএইচপি ব্যবহার করা যায়।
১৪. এটি সিকিউরড।
১৫. এটি ইউজার ফ্রেন্ডলি।
১৬. পিএইচপি এর Debugger engine দ্বারা খুব সহজে Error ডিটেক্ট করা যায়।
১৭. বেশির ভাগ সফটওয়্যার ফার্ম পিএইচপি Preferred করে।
১৮. পিএইচপি প্রায় সব ধরনের ডেটাবেজ ম্যানেজ করতে পারে।
১৯. এছাড়াও অবজেক্ট অরিয়েন্টেড ল্যাংগুয়েজের প্রায় সকল সুবিধা অবজেক্ট অরিয়েন্টেড পিএইচপিতে পাওয়া যাবে।

২.৬ অধ্যায় সারাংশ

> পিএইচপি একটি High Level ওপেন সোর্স স্ক্রিপ্টিং ল্যাংগুয়েজ। পিএইচপিকে বলা হয় হাইপারটেক্সট প্রিপ্রসেসর। এটি এক্সিকিউট করে ওয়েব সার্ভারে যেমন এ্যাপাচি অথবা আই আই এস এ।

অবজেক্ট অরিয়েন্টেড পিএইচপি

- > ক্লায়েন্ট কোন তথ্যর জন্য রিকোয়েস্ট করলে ওয়েব সার্ভারে যায়। পরে পিএইচপির মাধ্যমে সেটা প্রসেস হয় এরপর সে তথ্য ওয়েব সার্ভার ক্লায়েন্টকে পৌঁছে দেয়।
- > অবজেক্ট অরিয়েন্টেড পিএইচপি এর মাধ্যমে খুব সহজে ওয়েব বেইসড সফটওয়্যার এ্যাপ্লিকেশন তৈরি করা সম্ভব। এটির জন্য কোন লাইসেন্স ক্রয় করতে হয় না। এটি সিকিউরড ইউজার ফ্রেন্ডলি এবং লোডিং টাইম কম লাগে। ক্যারিয়ার হিসেবে রয়েছে এর ব্যাপক চাহিদা।

২.৮ প্রশ্ন

১. পিএইচপি কি?
২. পিএইচপি ল্যাংগুয়েজ কিভাবে কাজ করে বর্ণনা কর।
৩. অবজেক্ট অরিয়েন্টেড ল্যাংগুয়েজ পিএইচপি এর সুবিধা গুলো লিখ।
৪. ওয়েব সার্ভার কি?
৫. ক্লায়েন্ট রিকোয়েস্ট কিভাবে ওয়েব সার্ভার প্রসেস করে।


```
<?php  
echo "Hello Bangladesh";  
?>
```

Your First Programming Language

সহজ ভাষায় সবার জন্য...

পিএইচপি এন্ড মাইএসকিউএল

From Beginner to Professional



ওয়েব অ্যাপ্লিকেশন তৈরীতে বিশ্বের সর্বাধিক জনপ্রিয় ও ব্যবহৃত প্রোগ্রামিং ল্যাংগুয়েজ **PHP** ই হোক আপনার প্রথম পছন্দের ল্যাংগুয়েজ। নিজেকে প্রমাণ করুন একজন দক্ষ প্রোগ্রামার হিসেবে আর **PHP MySQL** এর সমন্বয়ে আপনিও তৈরী করুন বিভিন্ন ওয়েব ভিত্তিক অ্যাপ্লিকেশন ও ডাইনামিক ওয়েবসাইট।

১৩টি প্রজেক্ট সম্মিলিত

মোঃ মিজানুর রহমান

Learn Programming with Fun

With



Latest Version : PHP 5, MySQL 5

Download

www.bookbd.info

অধ্যায়-৩ পিএইচপি এনভায়নমেন্ট

এ অধ্যায়ের আলোচ্য বিষয় সমূহ

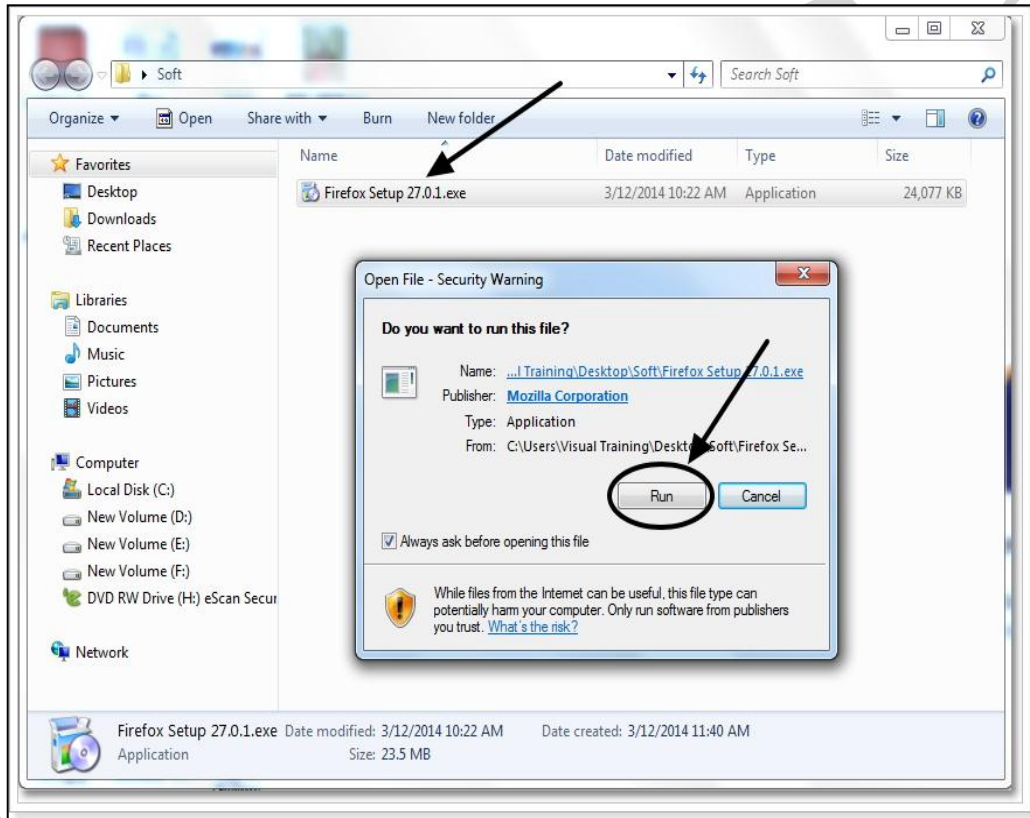
৩. ব্রাউজার এবং ওয়েব সার্ভার ইনস্টলেশন সম্পর্কে আলোচনা -----
- ৩.১ ভূমিকা -----
- ৩.২ ব্রাউজার ইনস্টলেশন -----
- ৩.৩ ওয়েব সার্ভার -----
- ৩.৪ ইনস্টল পিএইচপি -----
- ৩.৫ ইনস্টল ডেটাবেস মাইএসকিউএল -----
- ৩.৬ এ্যাপাচি (Apache) -----
- ৩.৭ আই আই এস (IIS) -----
- ৩.৮ জাম্প (XAMPP) -----
- ৩.৯ জাম্প ইনস্টল (XAMPP INSTALL) -----
- ৩.১০ ওয়াম্প (WAMP) -----
- ৩.১১ ওয়াম্প ইনস্টল -----
- ৩.১২ রান পিএইচপি প্রোগ্রাম -----

৩.১ ভূমিকাঃ

পিএইচপি প্রোগ্রামিং এর জন্য প্রথমেই আমাদের কম্পিউটারে পিএইচপির পরিবেশ তৈরী করে নিতে হবে। আমাদের কম্পিউটারে PHP, MySQL, Apache, ইত্যাদি ইনস্টল থাকতে হবে। আমরা উপরোক্ত সফটওয়্যার গুলো আলাদা আলাদা ইনস্টল করতে পারি অথবা একত্রে যে প্যাকেজ প্রোগ্রাম Xampp, Wamp, ইনস্টল করে নিতে পারি। আর কোড এর আউটপুট দেখার জন্য প্রয়োজন হবে ব্রাউজারের। তাই আপনার PC তে অবশ্যই ব্রাউজার থাকতে হবে।

৩.২ ব্রাউজার ইনস্টলেশনঃ

পিএইচপি কোড এর আউটপুট দেখতে হলে আমাদের প্রয়োজন হবে ওয়েব ব্রাউজারের। তাই আমরা এ অংশে দেখবো যে কিভাবে ওয়েব ব্রাউজার ইনস্টল করা যায়। এ জন্য প্রথমে বইয়ের সাথে যে সিডি দেওয়া আছে সেখানে Firefox setup 27.0.1.exe নামে যে ফাইলটি আছে তাতে ডাবল ক্লিক করুন। তাহলে নিচের মত মেসেজ আসবে। সেখানে Run এ ক্লিক করুন।



চিত্রঃ ৩.২.১ ব্রাউজার ইনস্টলেশন

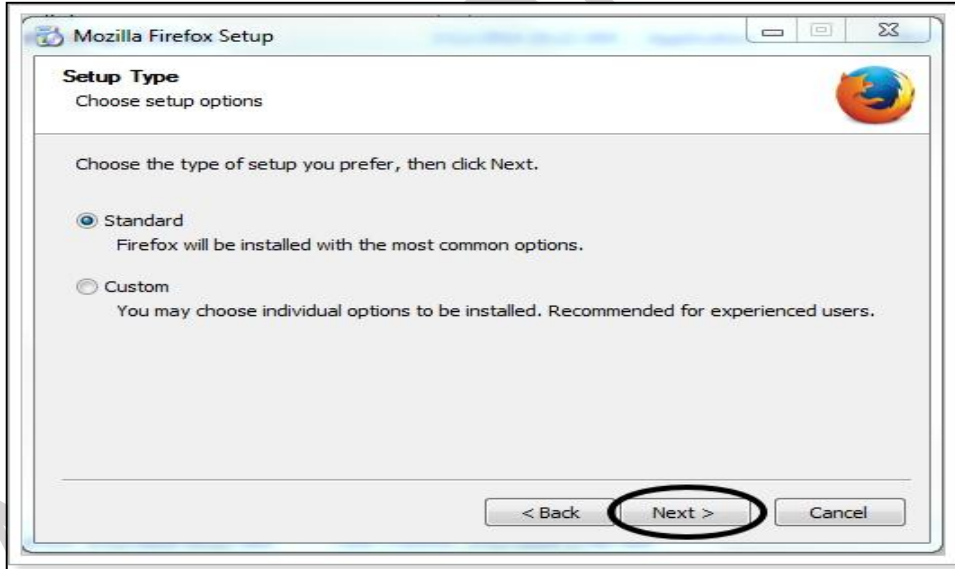
এরপর একটি মেসেজ আসবে সেখানে Yes এ ক্লিক করুন। তাহলে নিচের মত পেইজ আসবে। সেখানে Next এ ক্লিক করুন।

অবজেক্ট অরিয়েটেড পিএইচপি



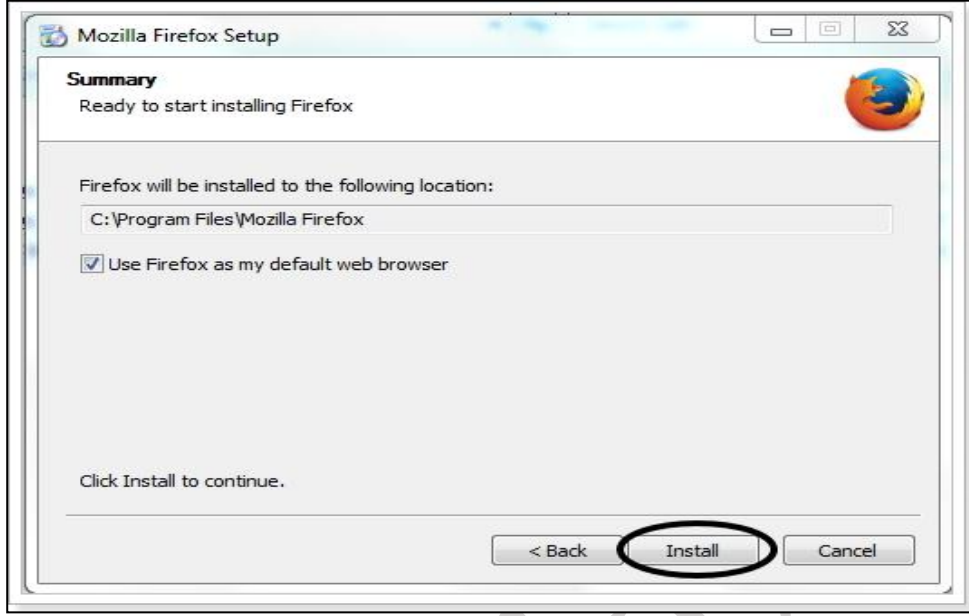
চিত্রঃ ৩.২.২ Next এ ক্লিক

তারপর আবার Next এ ক্লিক করুন।



চিত্রঃ ৩.২.৩ Next এ ক্লিক

এরপর যে পেইজটি আসবে সেখানে Install এ ক্লিক করুন।



চিত্রঃ ৩.২.৪ Install এ ক্লিক

এবার আপনার ব্রাউজারটি ইনস্টল সম্পূর্ণ হয়েছে।

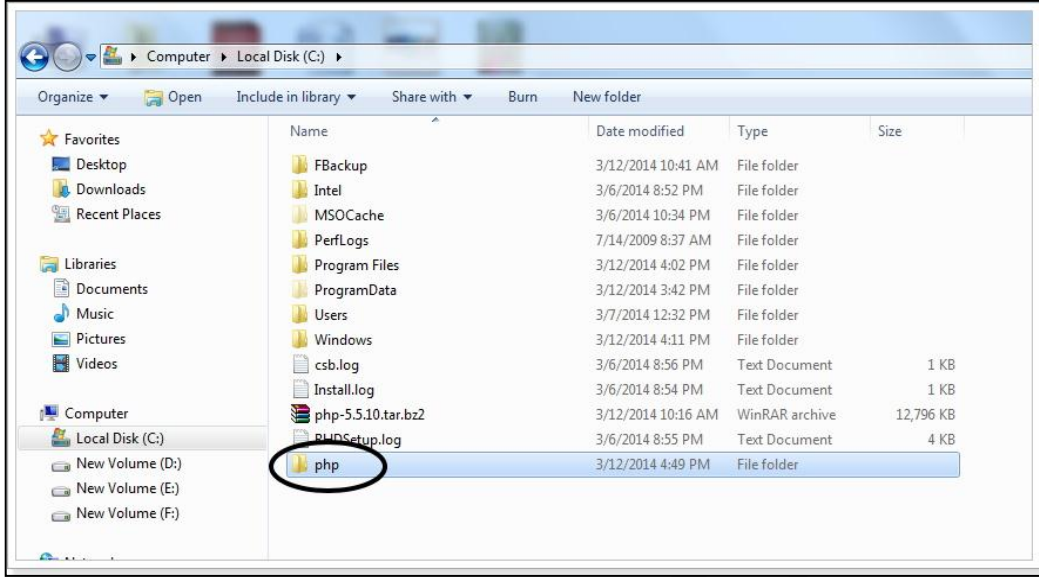
৩.৩ ওয়েব সার্ভারঃ

সার্ভার হলো এক ধরনের কম্পিউটার যেখানে ওয়েব পেইজ সংরক্ষিত থাকে। প্রত্যেকটি ওয়েব সার্ভারেরই একটি করে আইপি অ্যাড্রেস (IP Address) এবং থাকবে ডোমেইন নেইম। ধরুন আপনি www.bookbd.info অ্যাড্রেস আপনার ওয়েব ব্রাউজারে লিখে এন্টার প্রেস করেন তাহলে Bookbd.info সাইটে যত তথ্য দেওয়া আছে তা পরতে পারবেন। তো এই তথ্য কোন এক সার্ভারে সংরক্ষিত আছে। যেখানে সংরক্ষিত আছে সেটাই সার্ভার। আর www.bookbd.info হলো domain name. পিএইচপি হলো একটি সার্ভার সাইড স্ক্রিপটিং ল্যাংগুয়েজ। তাই এর কোড রান করতে হলে অবশ্যই সার্ভার প্রয়োজন হবে। পিএইচপিতে কাজ করার শুরুতেই আপনার কম্পিউটারে সার্ভার তৈরী করতে হবে। আর সে জন্য প্রয়োজন কিছু সফটওয়্যার ইনস্টল করা।

৩.৪ ইনস্টল পিএইচপি

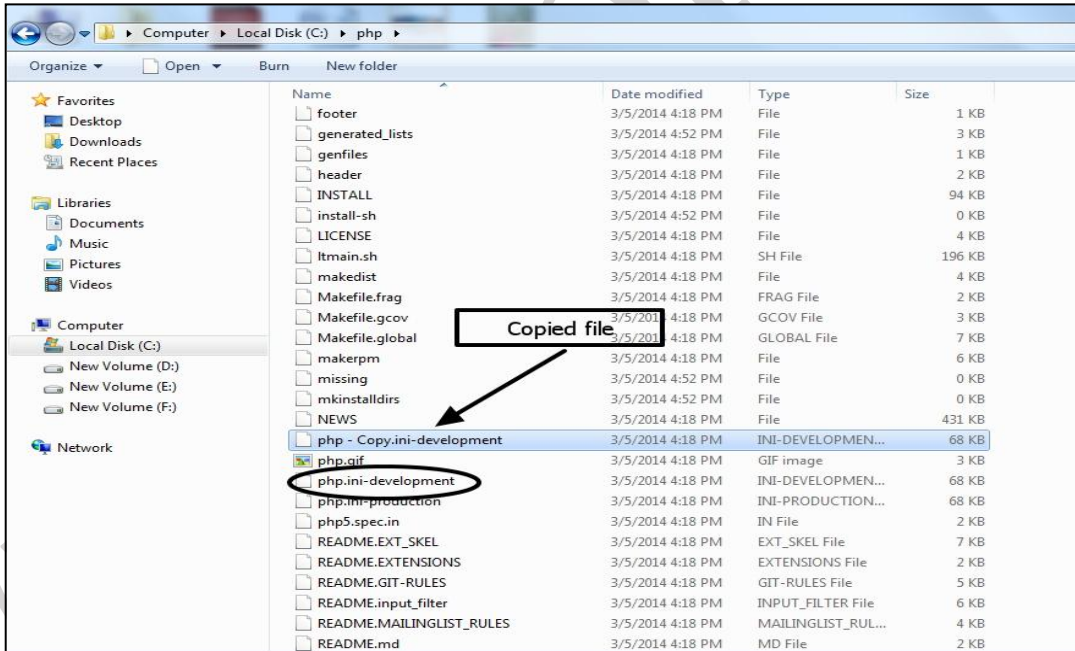
পিএইচপি ইনস্টল করার শুরুতে প্রথমেই আমাদের পিএইচপি সফটওয়্যার এর লেটেস্ট ভার্সন ডাউনলোড করে নিতে হবে এজন্য php.net/download.php থেকে ডাউনলোড করে নিন। অথবা আমাদের CD তে PHP এর ৫.৫.১০ ভার্সন টা দেওয়া আছে সেটা আপনার কম্পিউটারের C ড্রাইভে কপি করে নিন। তারপর ফাইলটাকে Extract করুন। Extract করার পর ফোল্ডারটির নাম পরিবর্তন করে PHP করে দিন। এবার PHP ফোল্ডারের ভিতরে গিয়ে `php.ini-development` ফাইলটাকে কপি করে এখানেই Paste করুন।

অবজেক্ট অরিয়েটেড পিএইচপি



চিত্রঃ ৩.৪.১

এবার PHP –copy.ini-development ফাইলটি নাম পরিবর্তন করে php.ini করে দিন।



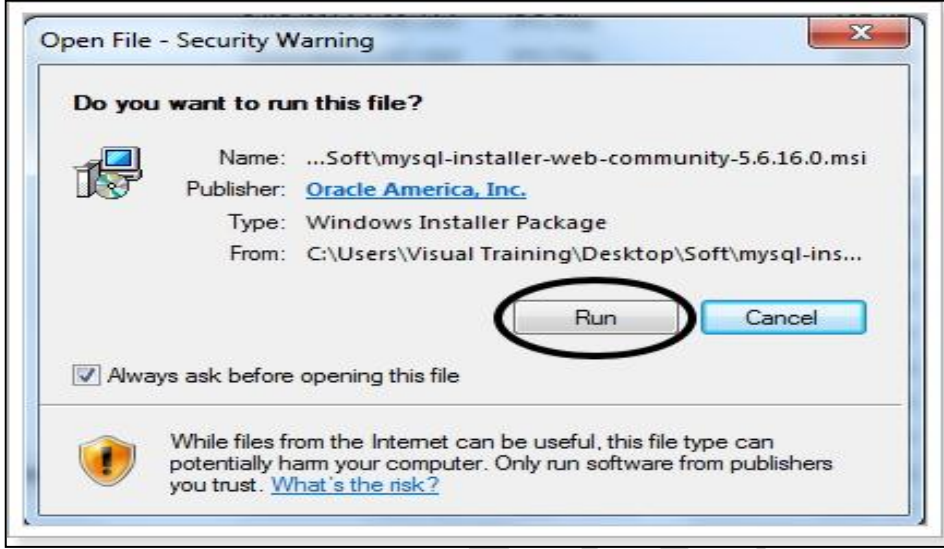
চিত্রঃ ৩.৪.২ php.ini ফাইল তৈরী

৩.৫ ইনস্টল ডেটাবেস মাইএসকিউএল

ডেটাবেজ তৈরীর জন্য আমাদের মাইএসকিউএল প্রয়োজন হবে। তাই এই অংশে আমরা দেখবো যে কিভাবে মাইএসকিউএল সফটওয়্যারটি ইনস্টল করা যায়। এ জন্য প্রথমে Mysql.com/download থেকে Mysql

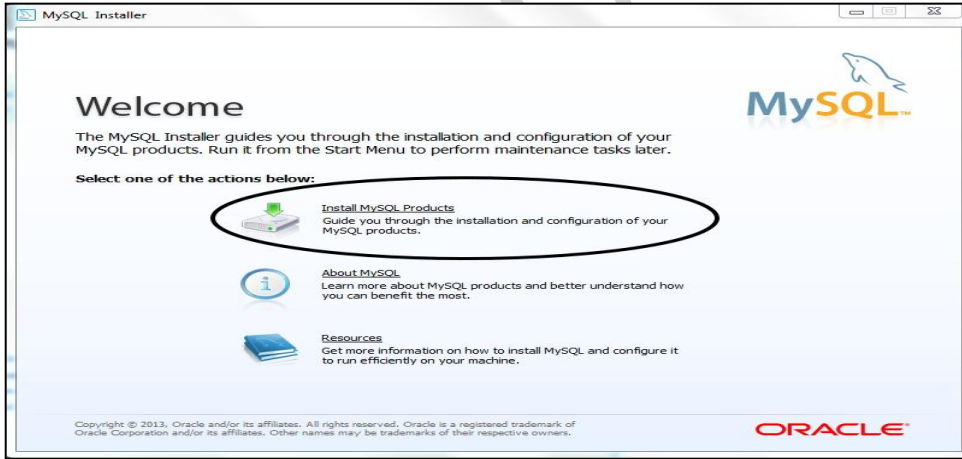
অবজেক্ট অরিয়েটেড পিএইচপি

ডাউনলোড করে নেই অথবা আমাদের CD তে Mysql সফটওয়্যারটি দেওয়া আছে। চাইলে সেটিও ব্যবহার করতে পারেন। প্রথমে CD থেকে Mysql-install-web-community-5.6.6.0 টাকে কপি করে আপনার কম্পিউটারে রাখুন। তারপর সেটাতে ডাবল ক্লিক করুন তাহলে নিচের মত একটি ডায়ালগ বক্স আসবে।



চিত্রঃ ৩.৫.১ Mysql ইনস্টল

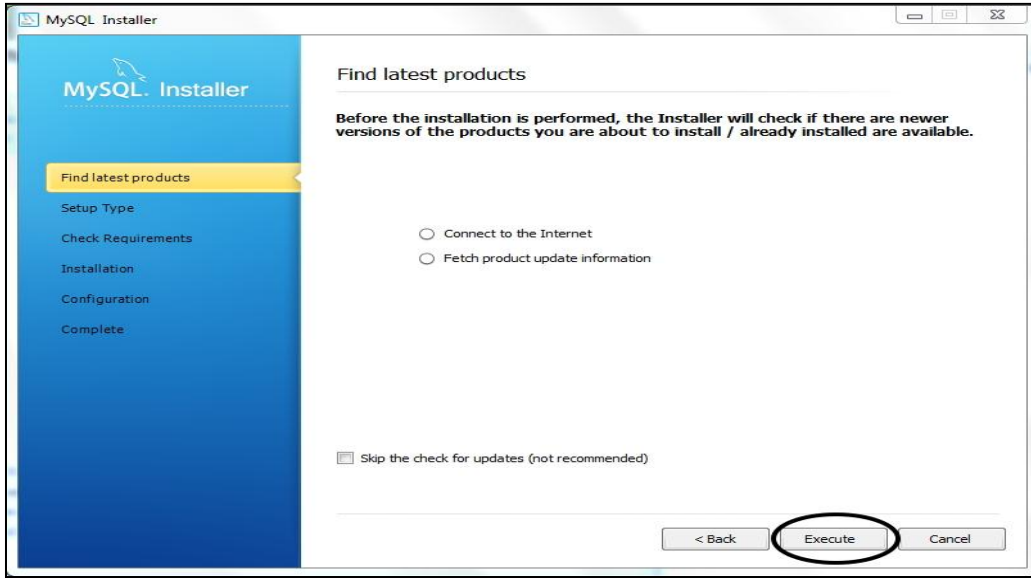
একটি ডায়ালগ বক্স আসবে সেখানে Yes এ ক্লিক করুন। তাহলে নিচের মত পেইজ আসবে। সেখান থেকে Install Mysql products এ ক্লিক করুন।



চিত্রঃ ৩.৫.২ Mysql ইনস্টল

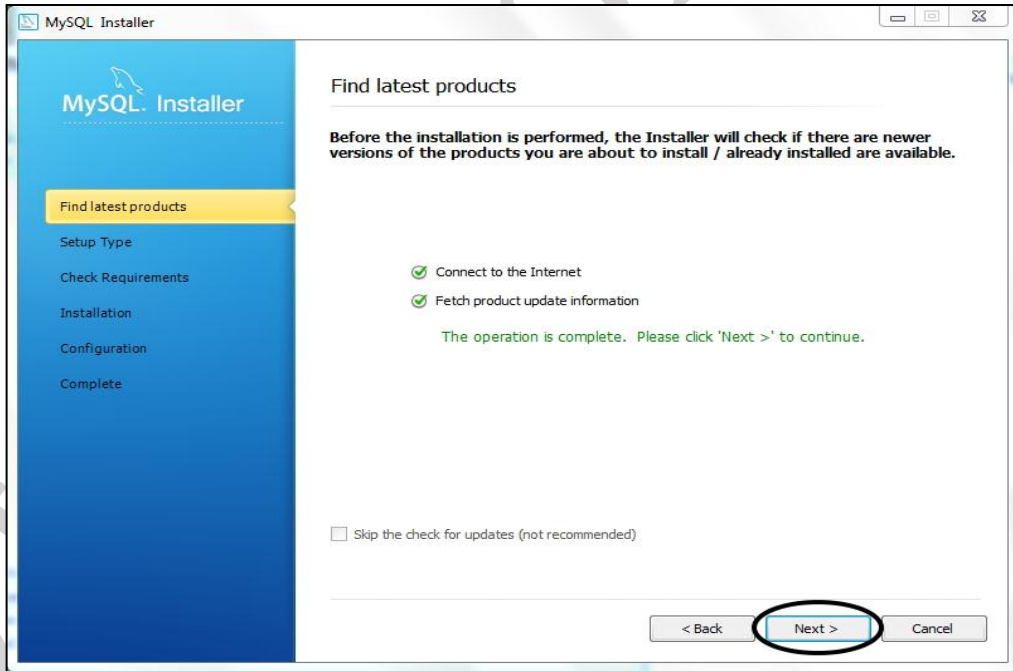
এবার Execute এ ক্লিক করুন তার আগে আপনার কম্পিউটারে ইন্টারনেট সংযোগ দিয়ে নিন।

অবজেক্ট অরিয়েটেড পিএইচপি



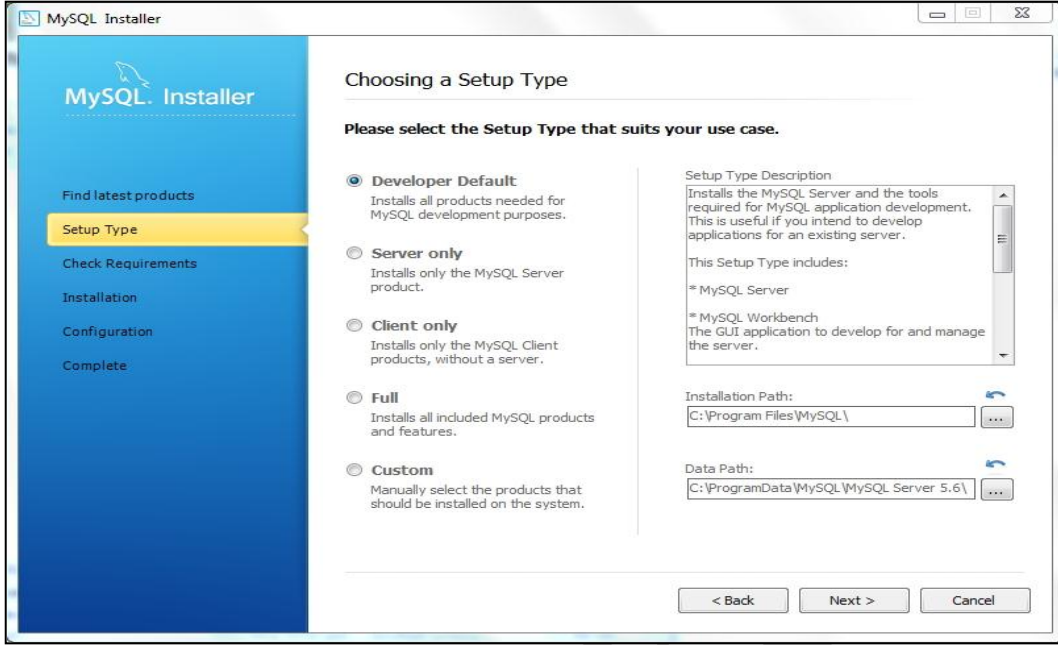
চিত্রঃ ৩.৫.৩

এবার সবুজ রংয়ের একটি মেসেজ প্রদর্শিত হবে। The Operation is Complete please click Next to continue.



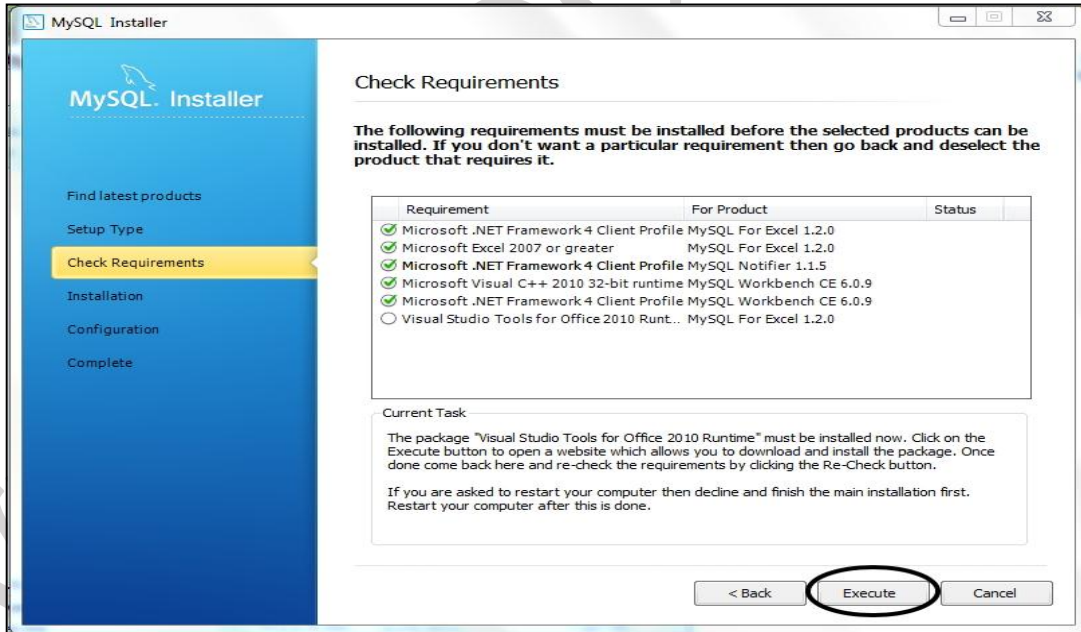
চিত্রঃ ৩.৫.৪

এরপর নিচের পেইজটি আসবে। সেখানে Developer Default সিলেক্ট রেখে Next এ ক্লিক করুন।



চিত্রঃ ৩.৫.৫

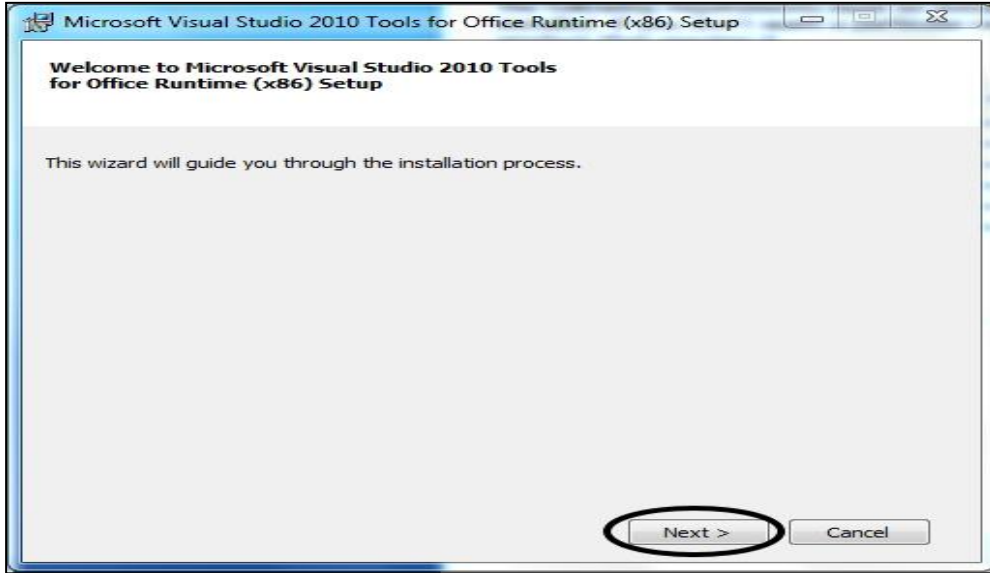
এরপর নিচের পেইজটি আসবে সেখানে Execute এ ক্লিক করুন।



চিত্রঃ ৩.৫.৬

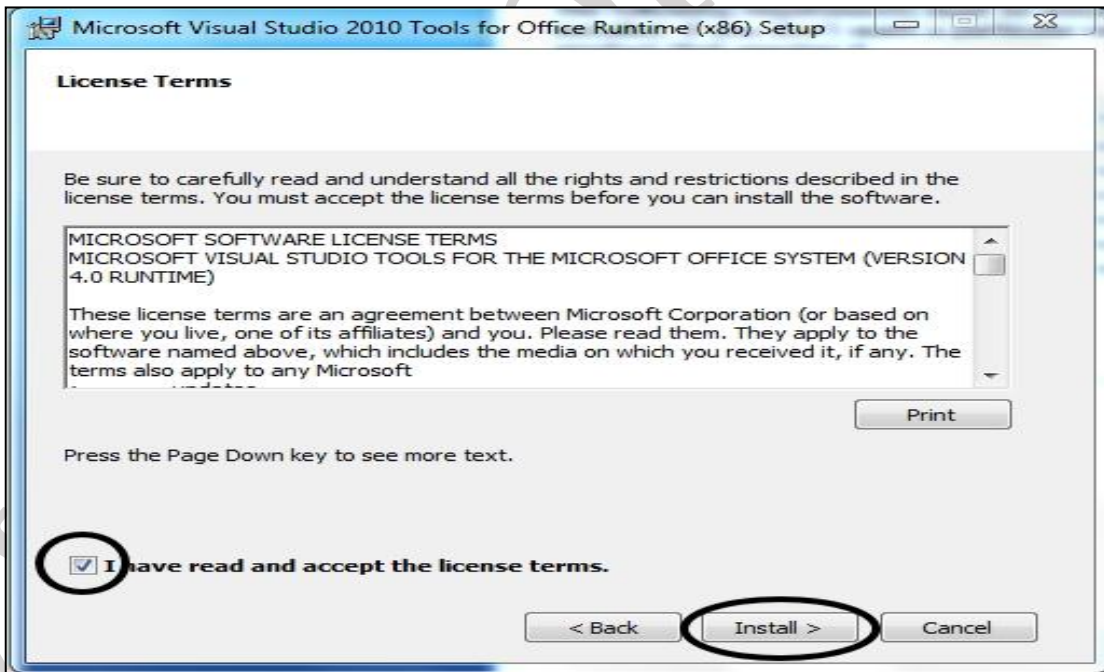
তারপর একটি ডায়ালগ বক্স আসবে সেখানে Next এ ক্লিক করুন।

অবজেক্ট অরিয়েটেড পিএইচপি



চিত্রঃ ৩.৫.৭

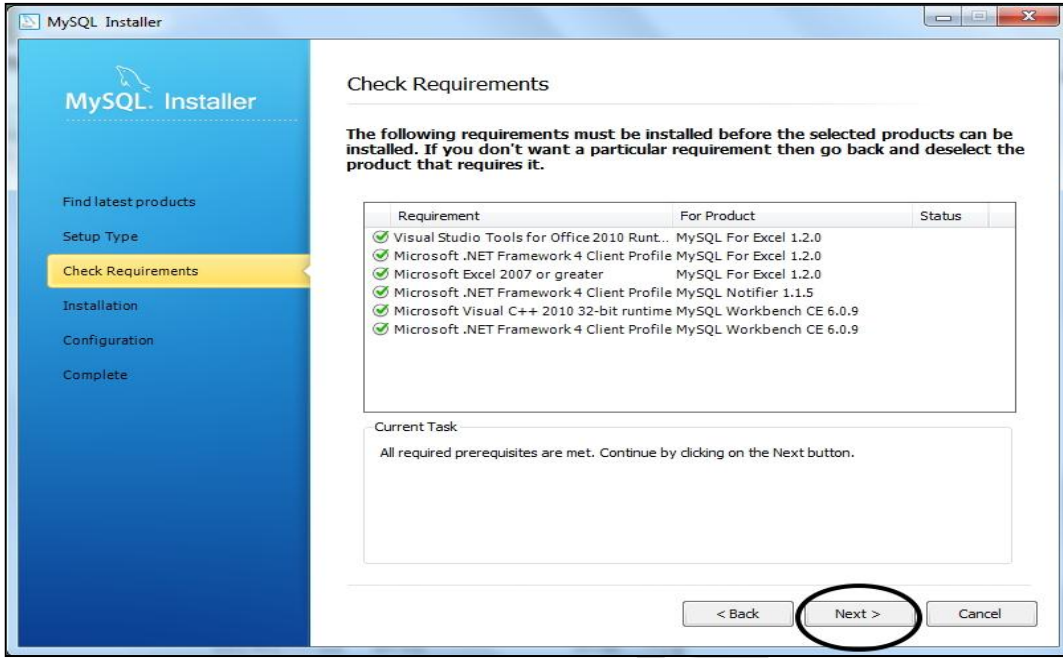
এরপর যে ডায়ালগ বক্স আসবে সেখানে I Have read and accept the license terms. টা চেকঅন করে install এ ক্লিক করুন।



চিত্রঃ ৩.৫.৮

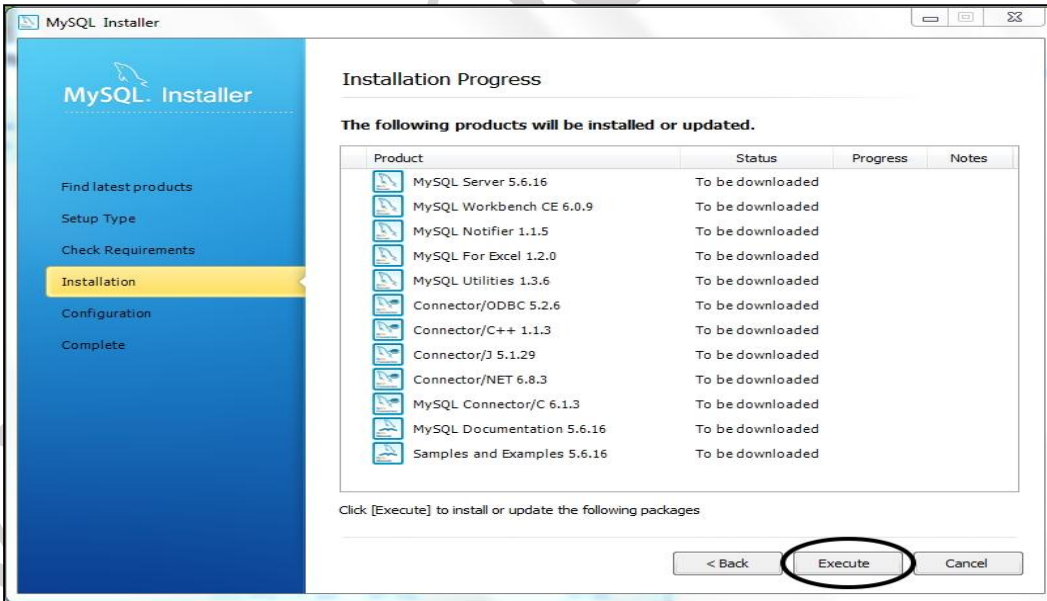
এবার Next এ ক্লিক করুন।

অবজেক্ট অরিয়েটেড পিএইচপি



চিত্রঃ ৩.৫.৯

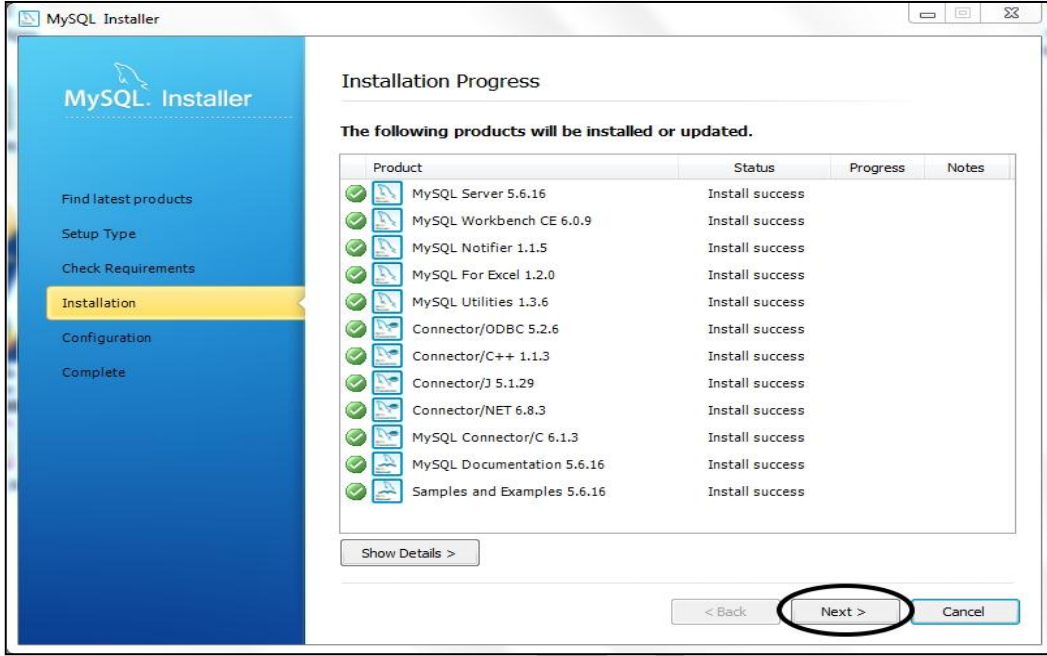
এবার Execute এ ক্লিক করুন।



চিত্রঃ ৩.৫.১০

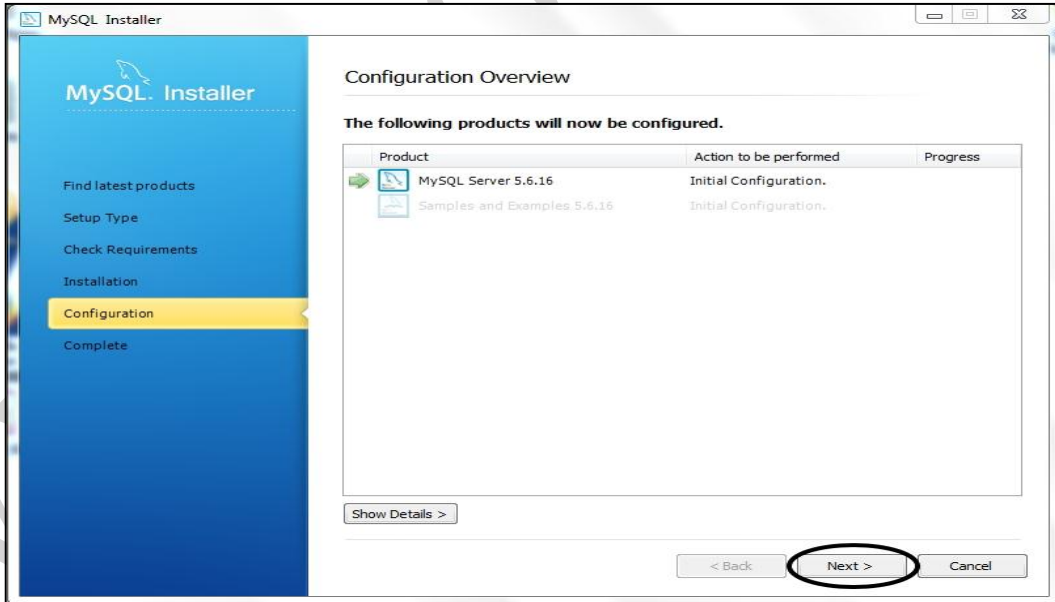
এবার প্রোডাক্ট গুলো ডাউনলোড হওয়া পর্যন্ত অপেক্ষা করুন। এখানে বেশ কিছুক্ষন সময় লাগবে। ডাউনলোড হয়ে গেলে install হবে। তারপর

অবজেক্ট অরিয়েটেড পিএইচপি



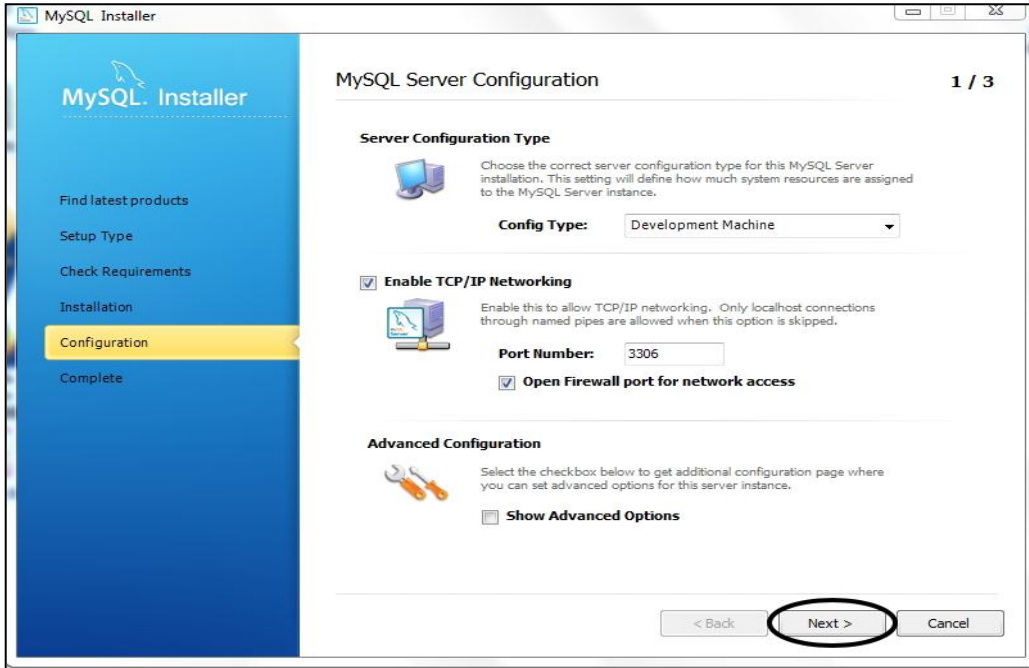
চিত্রঃ ৩.৫.১১

এবার যে পেইজটি আসবে সেখানে আবার Next এ ক্লিক করুন।



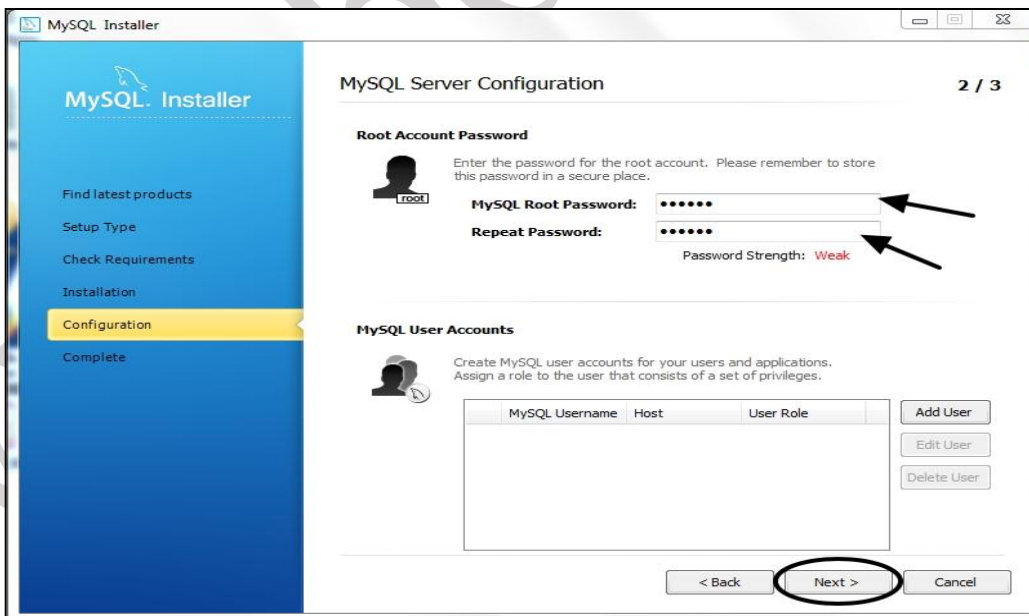
চিত্রঃ ৩.৫.১২

আবার Next এ ক্লিক করুন।



চিত্রঃ ৩.৫.১৩

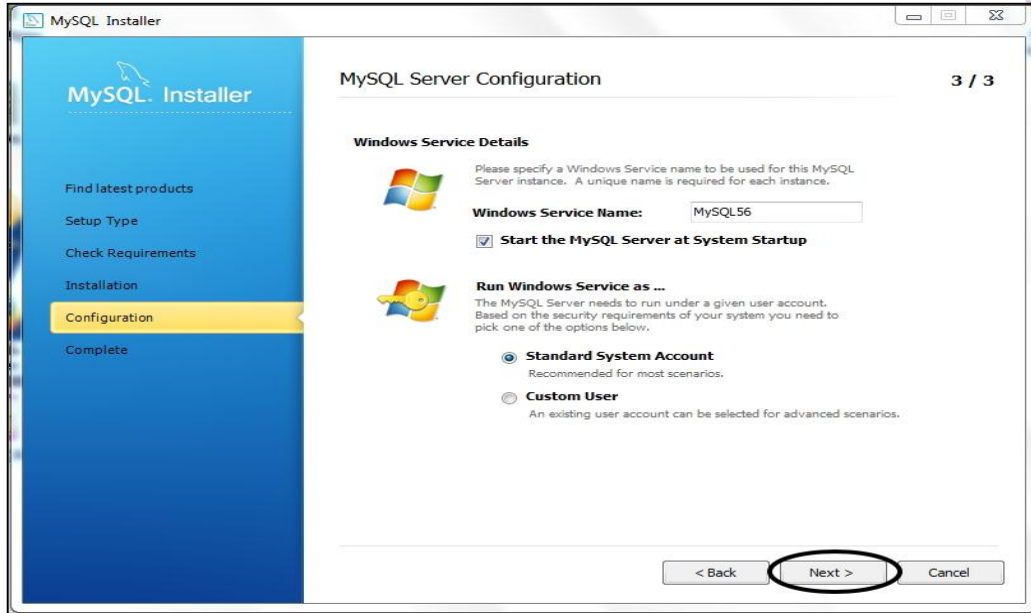
এবার এখানে একটি Password দিন এবং next এ ক্লিক করুন।



চিত্রঃ ৩.৫.১৪

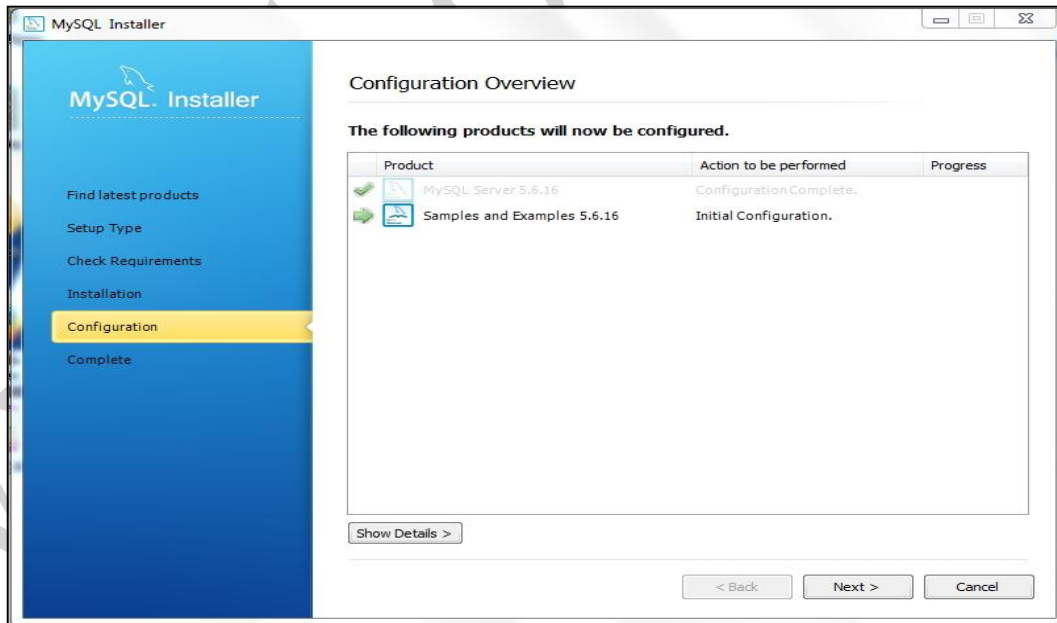
অবজেক্ট অরিয়েটেড পিএইচপি

এবার আবার next এ ক্লিক করুন।



৩.৫.১৫

এবার আবার next বাটনে ক্লিক করুন।

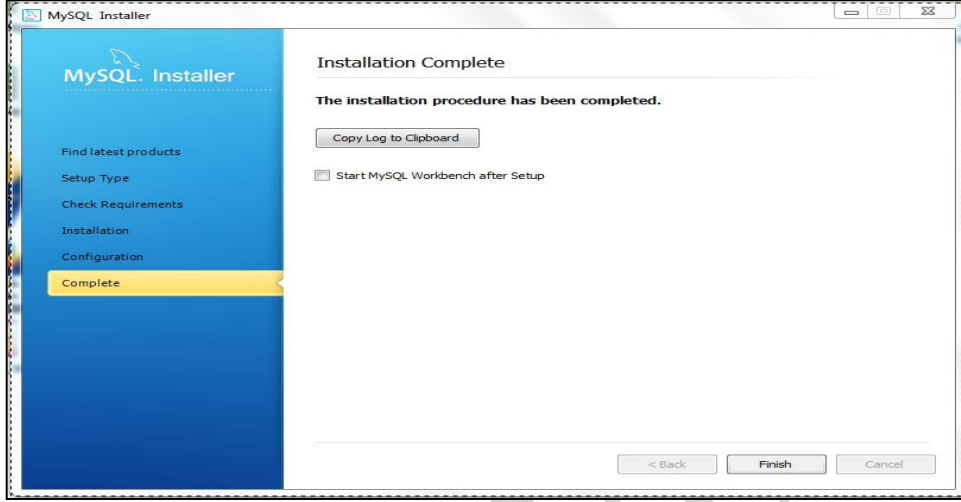


চিত্রঃ ৩.৫.১৬

এবার যে পেইজটি আসবে সেখানে আবার next এ ক্লিক করুন।

অবজেক্ট অরিয়েটেড পিএইচপি

এবার সবশেষে finish এ ক্লিক করুন।



চিত্রঃ ৩.৫.১৮

৩.৬ জাম্প (XAMPP)

Xampp হলো একটি ওপেন সোর্স প্যাকেজ প্রোগ্রাম। খুব সহজেই কম্পিউটারে লোকাল সার্ভার তৈরী করতে Xampp ব্যবহার করা হয়ে থাকে। Xampp এ Apache, mysql, php, perl ইত্যাদি একত্রে দেওয়া থাকে তাই কাজ করতে খুব সুবিধা হয়। এটি দিয়ে Windows, Mac, Linux এ কাজ করা যায়।

৩.৭ জাম্প ইনস্টল (XAMPP INSTALL)

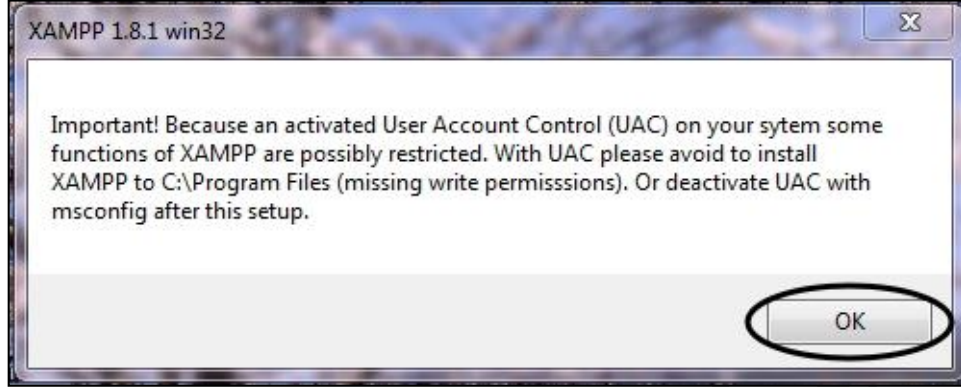
Xampp সফটওয়্যারটি আমরা প্রথমে www থেকে ডাউনলোড করে নিই অথবা আমাদের CD এর Software ফোল্ডারের ভিতর থেকে Xampp win-32-1.8.1-vc9-install টি কপি করে আপনার হার্ড ড্রাইভে রাখুন। তারপর Xampp-win32-1.8.1-vc9 install টিতে ডাবল ক্লিক করুন তাহলে একটি ডায়ালগ বক্স আসবে। সেখানে yes এ ক্লিক করুন। তাহলে নিচের মত মেসেজ আসবে। সেখানে ok তে ক্লিক করুন।



চিত্রঃ ৩.৯.১

অবজেক্ট অরিয়েটেড পিএইচপি

তারপর আবার একটি ডায়ালগ বক্স আসবে সেখানে ok তে ক্লিক করুন।



চিত্রঃ ৩.৯.২

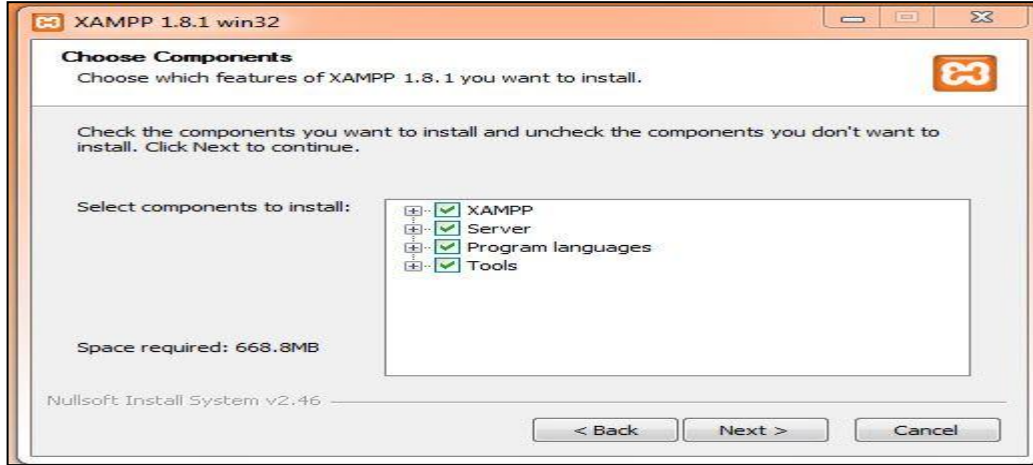
নিচের মত পেইজ আসবে সেখানে next এ ক্লিক করুন।



চিত্রঃ ৩.৯.৩ xampp installation

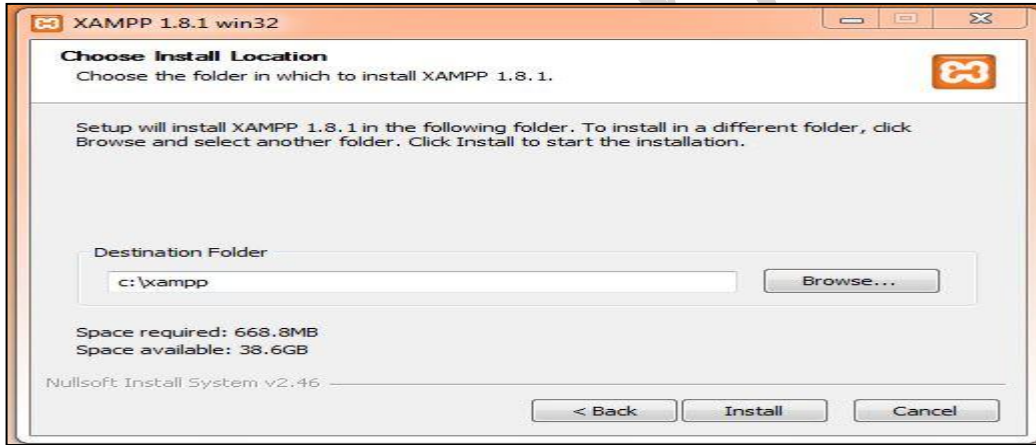
আবার নিচের মত পেইজ আসবে সেখানে next এ ক্লিক করুন।

অবজেক্ট অরিয়েটেড পিএইচপি



চিত্রঃ ৩.৯.৪

এবার আপনার Xampp কোন ডাইভে ইনস্টল করবেন তা এখানে নির্ধারন করে দিতে হবে। ডিফল্ট ভাবে C ড্রাইভে দেওয়া আছে। আমি C ড্রাইভে রেখেই ইনস্টল এ ক্লিক করেছি।



চিত্রঃ ৩.৯.৫

এবার ইনস্টল হওয়া পর্যন্ত কিছু সময় অপেক্ষা করুন। ইনস্টল হয়ে গেলে finish এ ক্লিক করুন। তারপর একটি ডায়ালগ বক্স আসবে সেখানে Yes এ ক্লিক করুন।

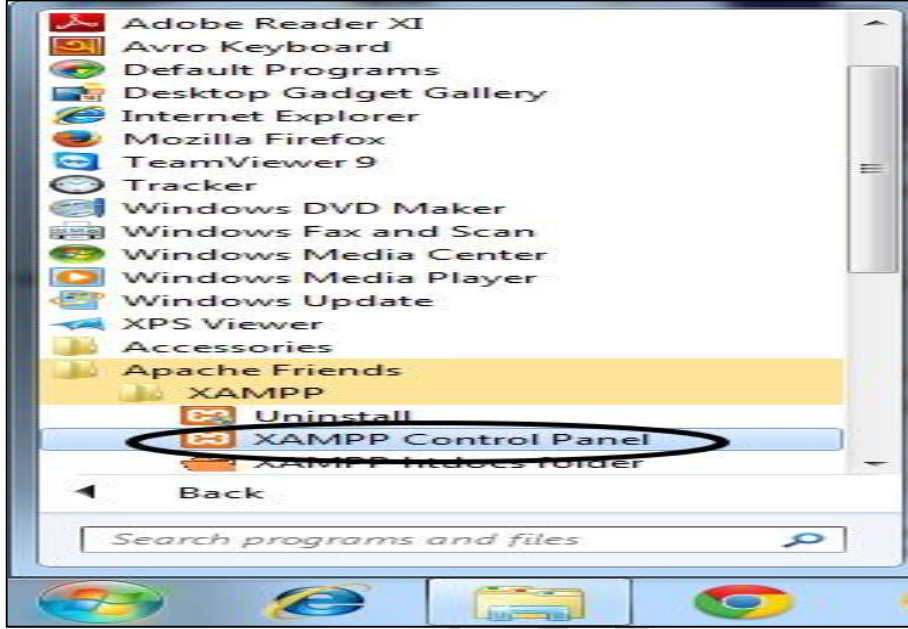


চিত্রঃ ৩.৯.৬

অবজেক্ট অরিয়েটেড পিএইচপি

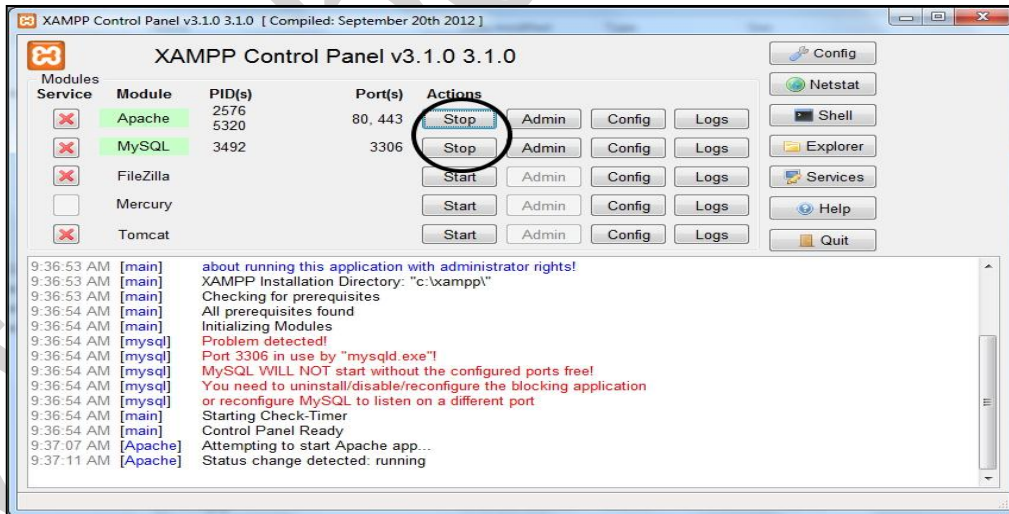
ব্যাস আপনার Xampp install হয়ে গেছে। এবার Xampp Control panel থেকে Apache এবং Mysql টি চালু করে দিন।

এবার Xampp ওপেন করার জন্য Start থেকে All Programs তারপর Apache friends তারপর Xampp Control panel এ ক্লিক করুন।



চিত্রঃ ৩.৯.৭

তাহলে নিচের পেইজটি আসবে সেখানে Apache এবং Mysql টা Start দিন।

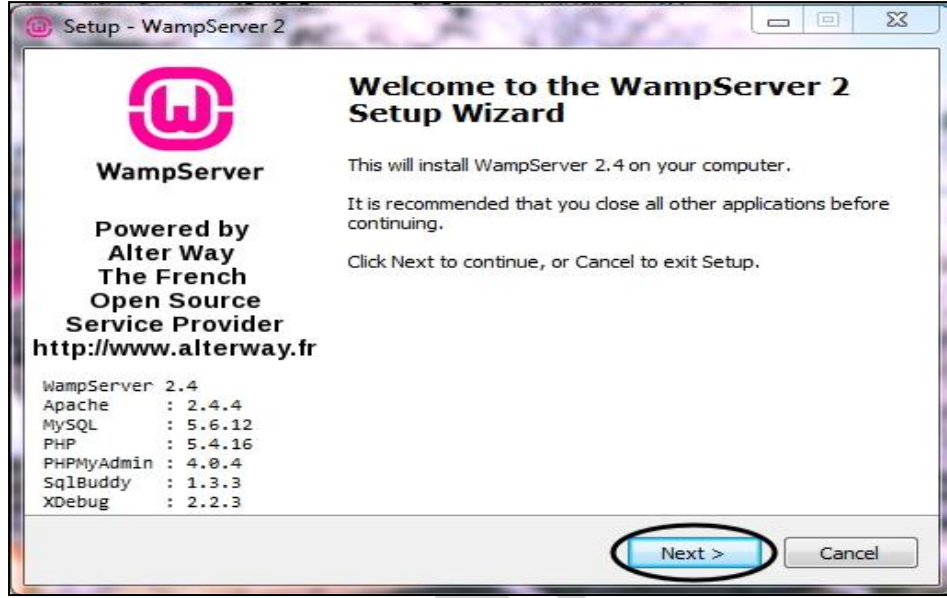


চিত্রঃ ৩.৯.৮

অবজেক্ট অরিয়েটেড পিএইচপি

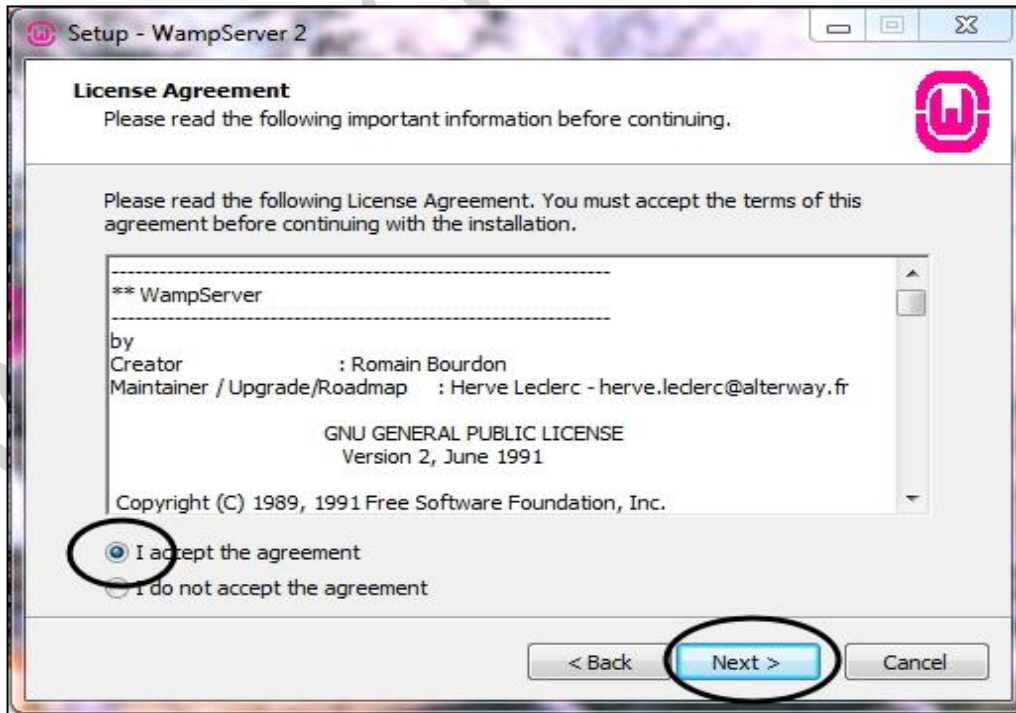
৩.৮ ওয়াম্প ইনস্টল

প্রথমে CD এর software folder থেকে wamps 2.4-X86 টাকে কপি করে আপনার হার্ডড্রাইভে রাখুন। তারপর এটিতে ডাবল ক্লিক করুন। তাহলে একটি পেইজ আসবে। সেখানে yes এ ক্লিক করুন। এবার নিচের মত পেইজ আসবে। সেখানে next এ ক্লিক করুন।



চিত্রঃ ৩.১১.১

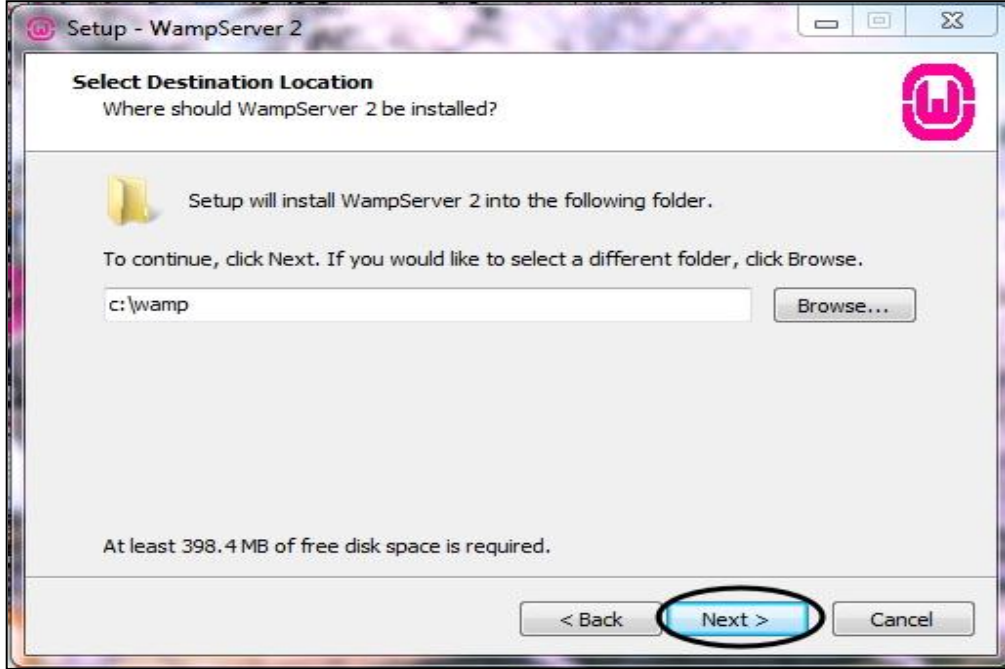
তারপর I Accept the agreement টাতে চেক অন করে তারপর next এ ক্লিক করুন।



চিত্রঃ ৩.১১.২

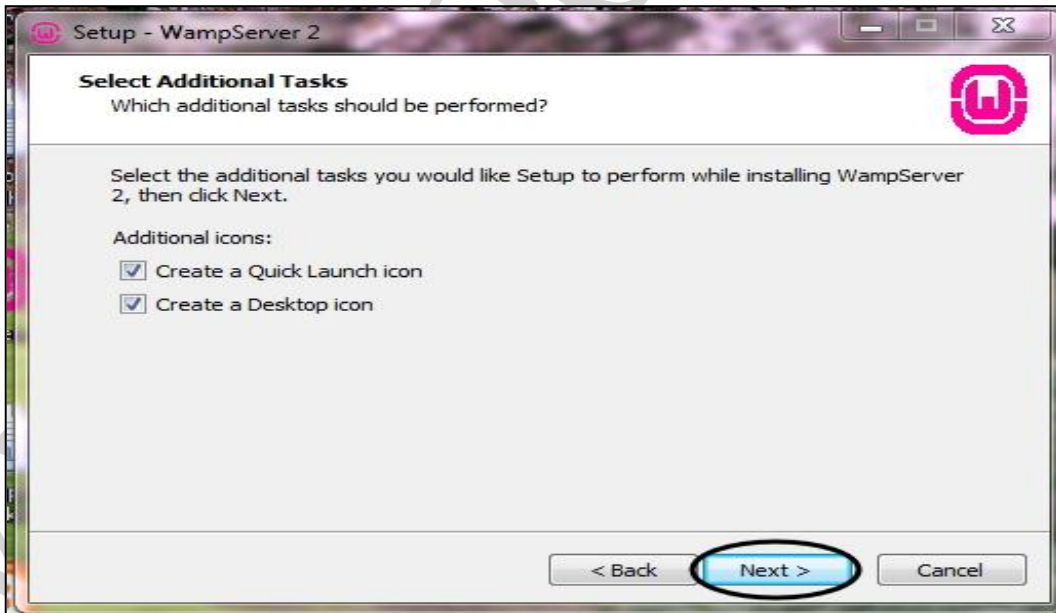
অবজেক্ট অরিয়েটেড পিএইচপি

এখানে আপনি কোন ড্রাইভে wamp ইনস্টল করতে চান তা নির্ধারণ করে দিন। তারপর next এ ক্লিক করুন।



চিত্রঃ ৩.১১.৩

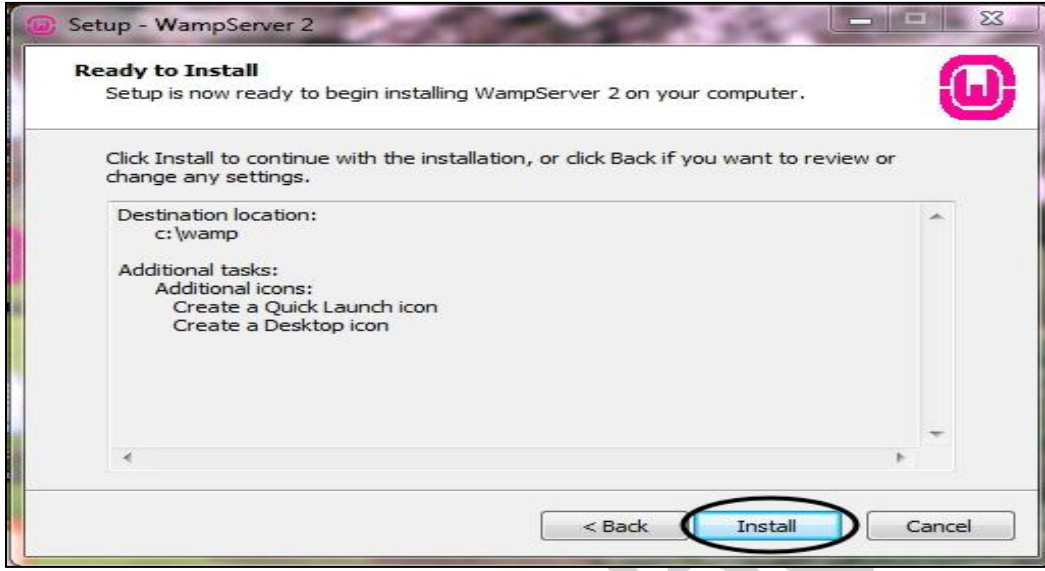
তারপর Create a quick launch icon এবং Create a desktop এটি চেক অন করে next এ ক্লিক করুন।



চিত্রঃ ৩.১১.৪

অবজেক্ট অরিয়েটেড পিএইচপি

এবার ইনস্টল এ ক্লিক করুন।



চিত্রঃ ৩.১১.৫

এবার কিছুক্ষন অপেক্ষা করুন। তারপর নিচের যে পেইজটি আসবে সেখানে email এর ঘরে আপনার ইমেইলটি দিয়ে next এ ক্লিক করুন।



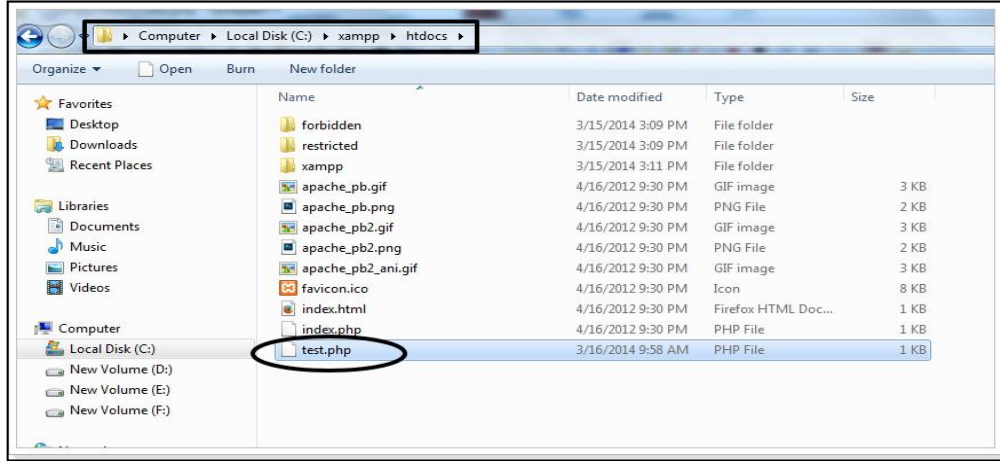
চিত্রঃ ৩.১১.৬

এবার finish এ ক্লিক করুন। ব্যাস আপনার Wampp Installation সম্পূর্ণ হয়েছে।

৩.১২ রান পিএইচপি প্রোগ্রাম

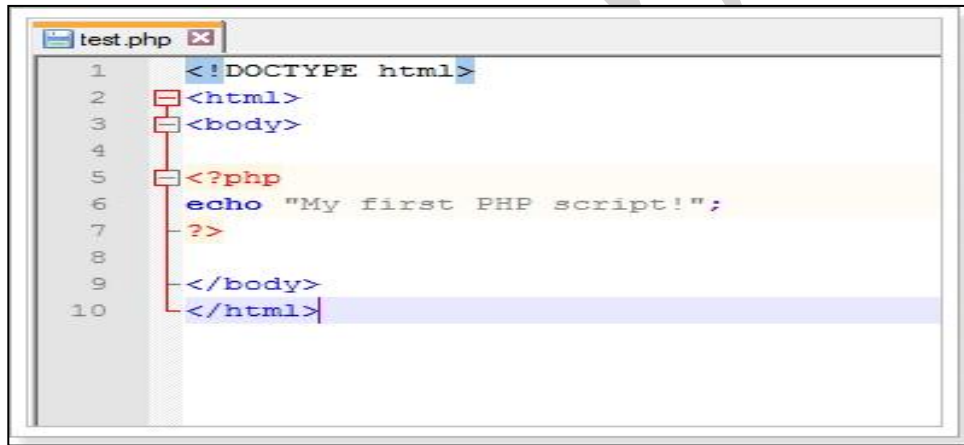
প্রথমে আপনি আপনার C Drive তারপর Xampp তারপর htdocs ফোল্ডারের ভিতর text.php নামে একটি file তৈরী করুন।

অবজেক্ট অরিয়েটেড পিএইচপি



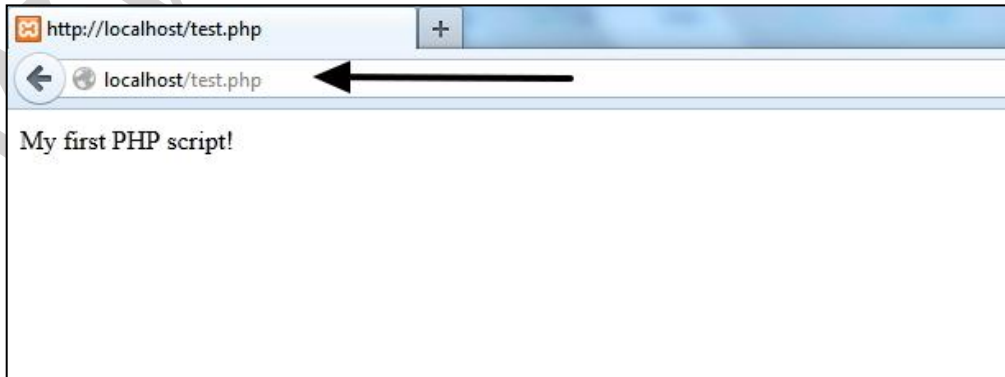
চিত্রঃ ৩.১২.১

এবার যে কোন একটি টেক্সট এডিটর দিয়ে test.php নামের ফাইলটি ওপেন করুন এবং নিচের মত কোড লিখি এবং ফাইলটি Save করুন।



চিত্রঃ ৩.১২.২

এবার আপনার ব্রাউজারটি ওপেন করুন এবং Address বার এ Localhost/test.php লিখে এন্টার চাপুন।



চিত্রঃ ৩.১২.৩

অধ্যায়-৪ বেসিক অবজেক্ট অরিয়েন্টেড পিএইচপি

৪. পিএইচপি ফাইল, অবজেক্ট অরিয়েন্টেড পিএইচপি সিনট্যাক্স, পিএইচপি, এইচটিএমএল, ক্লাস, অবজেক্টের আলোচনা --

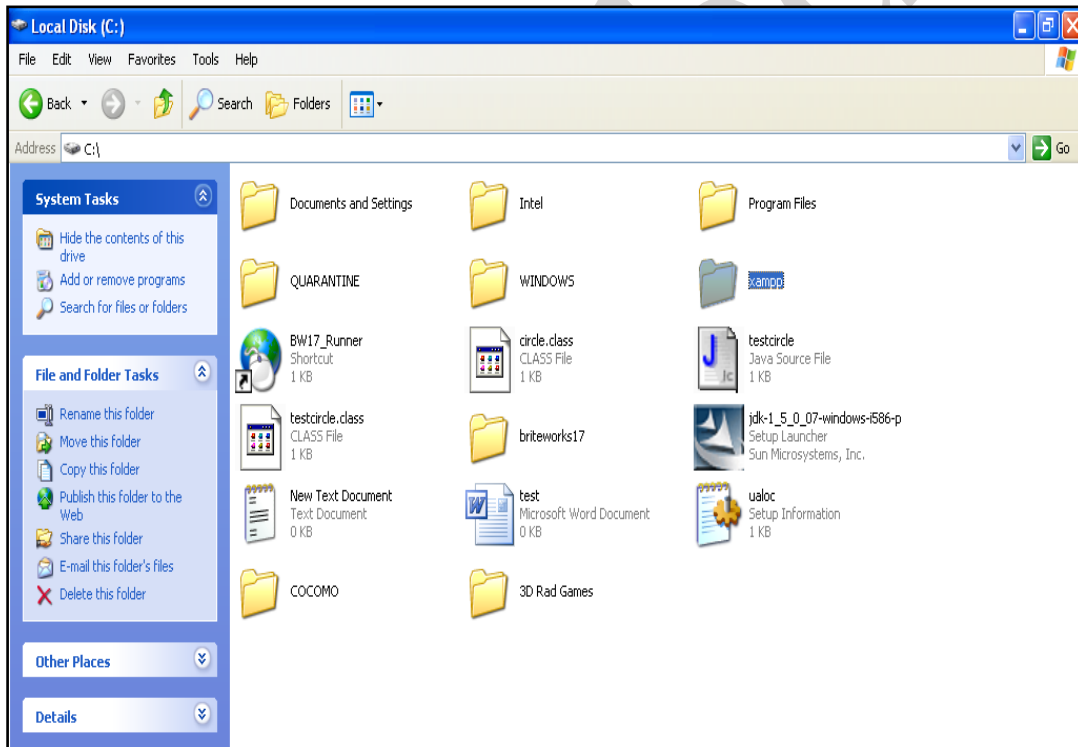
- ৪.১ ভূমিকা -----
- ৪.২ পিএইচপি ফাইল তৈরী -----
- ৪.৩ অবজেক্ট অরিয়েন্টেড পিএইচপি সিনট্যাক্স -----
- ৪.৪ ডিসপ্লে Hello Bangladesh -----
- ৪.৫ পিএইচপি এবং এইচটিএমএল -----
- ৪.৬ ক্লাস তৈরী -----
- ৪.৭ অবজেক্ট তৈরী -----
- ৪.৮ অবজেক্ট এবং ক্লাস -----

৪.১ ভূমিকা :

আমরা জানি পিএইচপি হচ্ছে সার্ভার সাইড স্ক্রিপ্টিং ল্যাংগুয়েজ তাই পিএইচপি প্রোগ্রাম রান করতে হলে আপনার কম্পিউটারে অবশ্যই অ্যাপাসি অথবা আই আই এস ওয়েব সার্ভার থাকতে হবে। আমরা সাধারণত Xampp অথবা Wamp ইত্যাদি সফটওয়্যার ইনস্টল করে থাকি। প্রতিটি ল্যাংগুয়েজের মত পিএইচপির ও রয়েছে নিজস্ব কিছু নিয়ম (সিনট্যাক্স)। এর অধ্যায় আমরা দেখবো কিভাবে পিএইচপি প্রোগ্রাম কম্পিউটারে রান করতে হয়।

৪.২ পিএইচপি ফাইল তৈরী

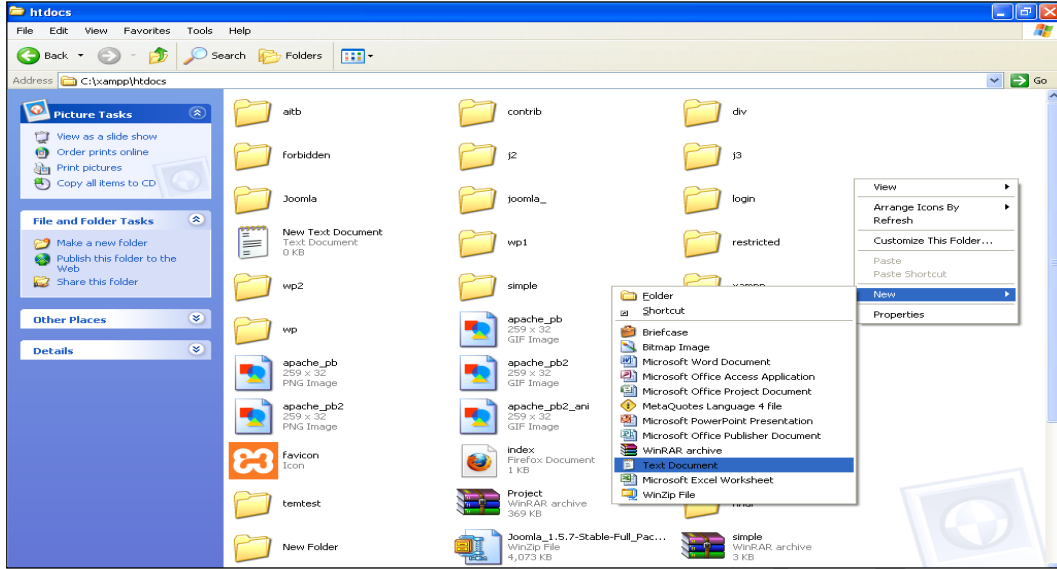
পি.এইচ.পি কোড লিখার জন্য আপনার একটি টেক্সট এডিটর দরকার, যেমন-উইডোজ নোটপ্যাড। এছাড়া, আপনি বিভিন্ন ধরনের IDE (Integrated Development Environment) Software যেমন:- (Macromedia Dreamweaver, Adobe Dreamweaver ইত্যাদি) ও ব্যবহার করতে পারেন, তবে এখানে আপনাদেরকে নোটপ্যাডে কোড লিখে দেখাবো। চলুন শুরু করা যাক। প্রথমে আপনি আপনার কম্পিউটারের যেখানে XAMPP (এছাড়াও আপনি অন্যান্য সফটওয়্যার যেমন:-WAMP) সফটওয়্যারটি ইনস্টল করেছেন, সেখানে চলে যান। এটি সাধারণত বাই ডিফল্ট C ড্রাইভে (System Drive) ইনস্টল হয়।



চিত্রঃ ৪.১ XAMPP ফোল্ডারটির physical অবস্থান

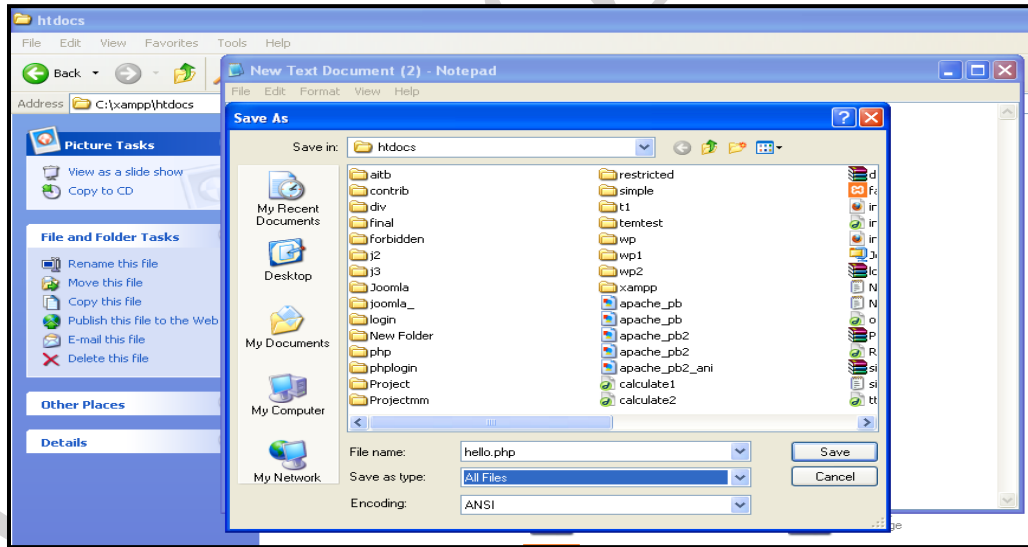
এখন আপনি XAMPP ফোল্ডারের ভিতর htdocs নামক একটি ফোল্ডার দেখতে পাবেন, এবার htdocs ফোল্ডারে প্রবেশ করুন। এখন htdocs এর ভিতর একটি Text Document ফাইল তৈরি করুন।

অবজেক্ট অরিয়েটেড পিএইচপি

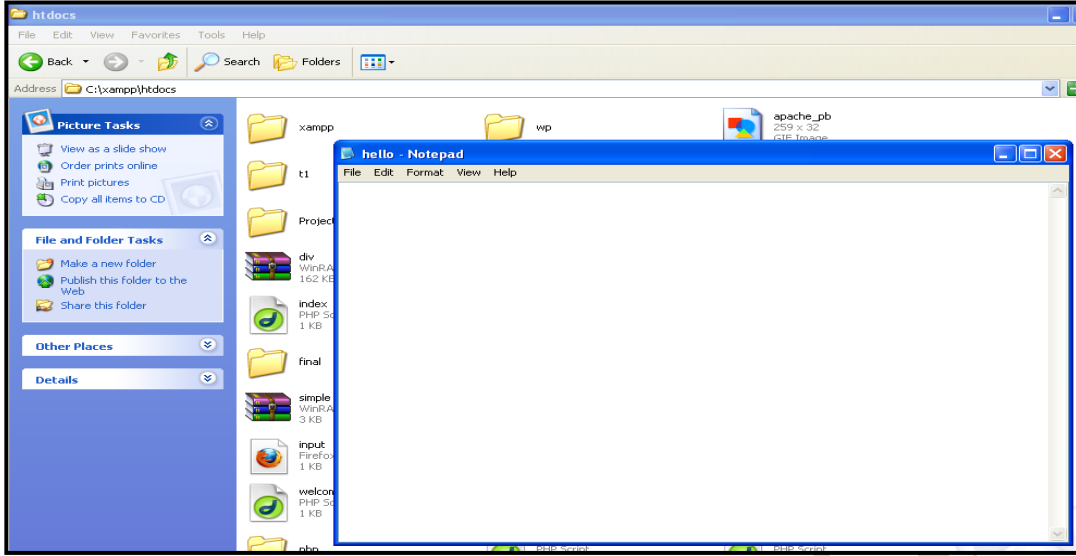


চিত্রঃ ৪.২ রাইট বাটন ক্লিক করে নতুন একটি Text Document তৈরি করা

এবার ফাইলটি ওপেন করে ফাইলটির ফাইল অপশন থেকে Save as এ ক্লিক করুন। সেখানে File Name এ ফাইলটির যে কোন একটি নাম দিন, যেমন আমি এখানে ফাইলটির নাম দিয়েছি hello.php। এখানে মনে রাখবেন, ফাইলের নাম অবশ্যই (.php) extension সহ লিখতে হবে, কারণ এটি PHP ফাইল লিখার নিয়ম (Syntax ও বলতে পারেন) এবং নিচে Save as Type এ All Files সিলেক্ট করে দিবেন।



চিত্রঃ ৪.৩ hello.php ফাইল নামকরণ



চিত্রঃ ৪.৫ hello.php ফাইল

উপরে লক্ষ্য করুন, hello.php নামক একটি ফাইল তৈরি হয়েছে।

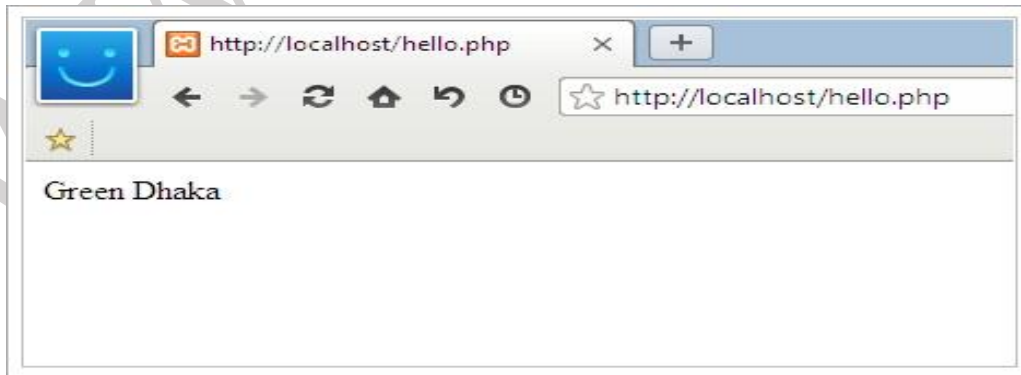
৪.৩ অবজেক্ট অরিয়েন্টেড পিএইচপি সিনট্যাক্সঃ

পিএইচপি ফাইলে পিএইচপি ট্যাগের (Tag) এর ভিতর সমস্ত পিএইচপি কোড লিখতে হবে।

<?php ----- ?> এই ট্যাগ এর ভিতর পিএইচপি কোড গুলো লিখবেন। আর এটিই হচ্ছে এক ধরনের পিএইচপি সিনট্যাক্স। ফাইল সেভ করতে হয় ফাইল এর নামদিয়ে . (dot) পিএইচপি (php) লিখতে হয়। নিম্নে একটি প্রোগ্রাম লিখছি যেটি আউটপুট হিসেবে ব্রাউজারে Green Dhaka প্রদর্শিত করবে।

```
<?php  
echo "Green Dhaka";  
?>
```

উপরের প্রোগ্রামে echo হচ্ছে একটি পিএইচপি কী-ওয়ার্ড এরপর ডাবল কোটেশন দিয়ে ভিতরে যা লিখবেন আউটপুট হিসেবে তা প্রদর্শিত করবে। নিম্নে চিত্রের মাধ্যমে দেখানো হল।



চিত্রঃ ৪.৩.১

8.8 ডিসপ্লে Hello Bangladesh :

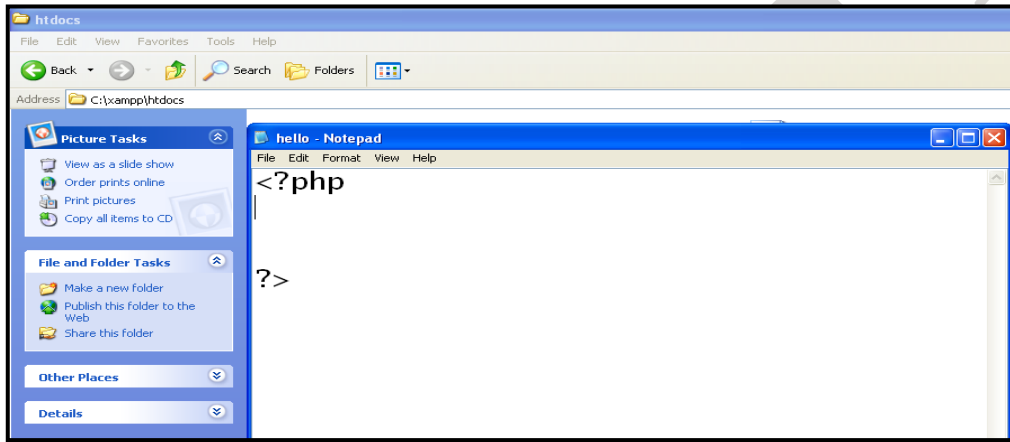
এতক্ষণ আপনাদের দেখালাম কিভাবে একটি php ফাইল তৈরি করতে হয়। এখন দেখাবো, কিভাবে ঐ পি.এইচ.পি ফাইলটিতে কোড লিখতে হবে। পি.এইচ.পি ফাইল এর সমস্ত কোড php syntax এর ভিতর লিখতে হয়। পি.এইচ.পি ফাইল কিভাবে শুরু এবং শেষ করতে হয় তার syntax নিচে দেওয়া হল :

প্রোগ্রাম নং-১ পি.এইচ.পি ফাইল শুরু এবং শেষ করার syntax

```
<?php
```

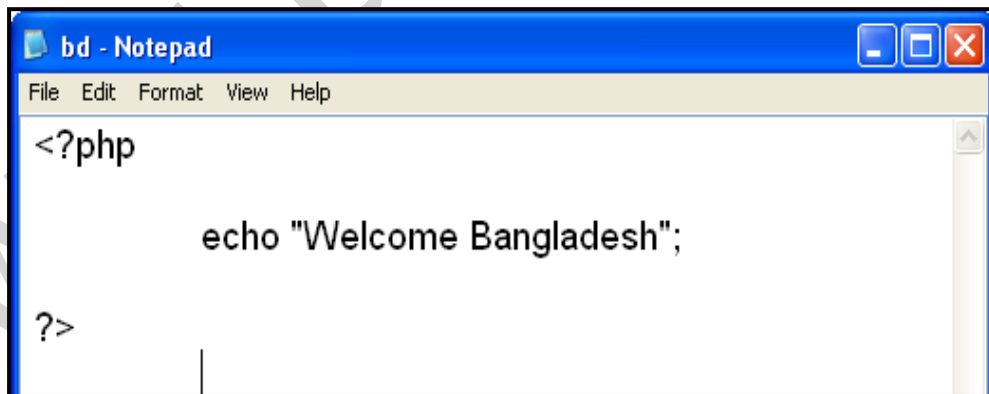
```
?>
```

পি.এইচ.পি অবশ্যই php স্ক্রিপ্ট এর মধ্যে লিখতে হবে। php script শুরু হয় <?php এবং শেষ হয় ?> দিয়ে।



চিত্র p.1.php: php open and close script

এখন আমরা এই script এর মধ্যে কোড লিখবো।

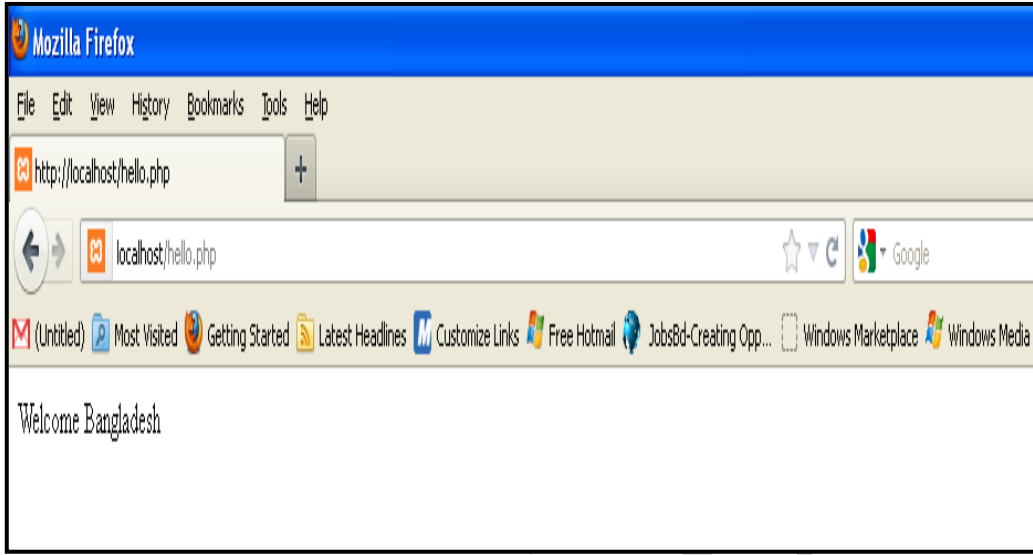


চিত্র (৩.২.২) : পি.এইচ.পি কোড

উপরের ফাইলটিতে লক্ষ্য করলে দেখবেন যে, আমরা আমাদের পি.এইচ.পি স্ক্রিপ্ট এর মধ্যে echo "Welcome Bangladesh"; লিখেছি। এখন আমরা এই ফাইলটি রান করতে চাচ্ছি। তবে php ফাইল রান করার পূর্বে অবশ্যই মনে রাখবেন, আপনার কম্পিউটারে Apache সার্ভার যেন চালু অবস্থায় থাকে। চলুন এখন দেখি, কিভাবে ফাইলটিকে

অবজেক্ট অরিয়েটেড পিএইচপি

রান করতে হয়। প্রথমে আমরা যে কোন একটি ব্রাউজার ওপেন করবো এবং ব্রাউজারের ইউআরএল (URL) বক্স এ লিখবো `http://localhost/hello.php` এবং লিখার পর কী-বোর্ড থেকে এন্টার (Enter) press করবো।

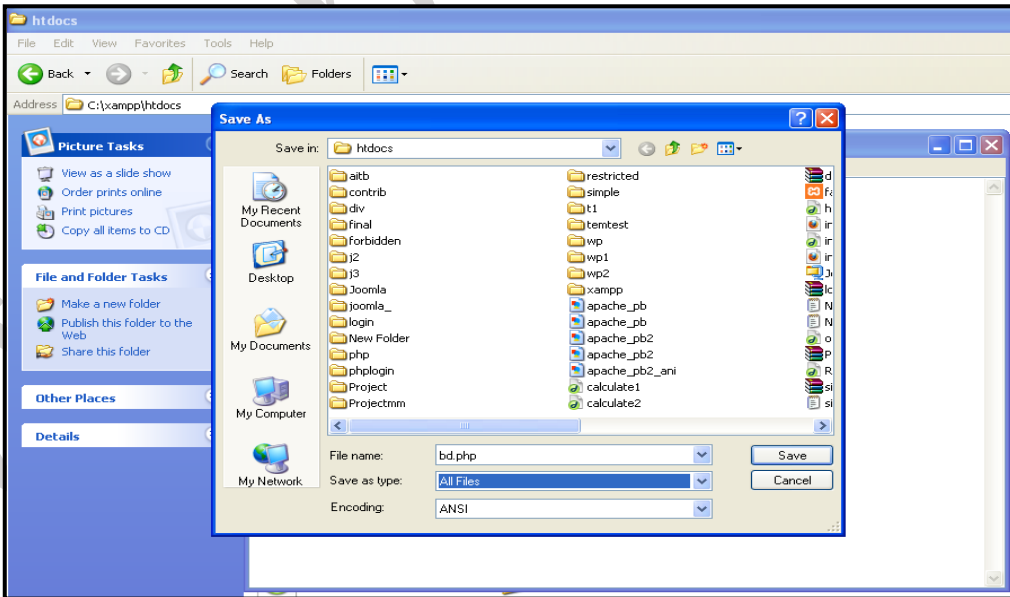


চিত্র ৩.২.৩ : hello.php ফাইল-এর আউটপুট

এখানে আউটপুট হিসেবে Welcome Bangladesh দেখাচ্ছে।

৪.৫ ইম্বেডেড পিএইচপি ফাইল :

আমরা এখন দেখবো, কিভাবে এইচ.টি.এম.এল ফাইলের ভিতর পি.এইচ.পি কোড লিখতে হয়। তবে অবশ্যই মনে রাখবেন, এখানেও আপনার ফাইলটির এক্সটেনশন (.php) হবে। আবারো একটি ফাইল ওপেন করলাম এবং ফাইলটির নাম দিচ্ছি hello.php এবং Save as type থেকে All Files সিলেক্ট করলাম।



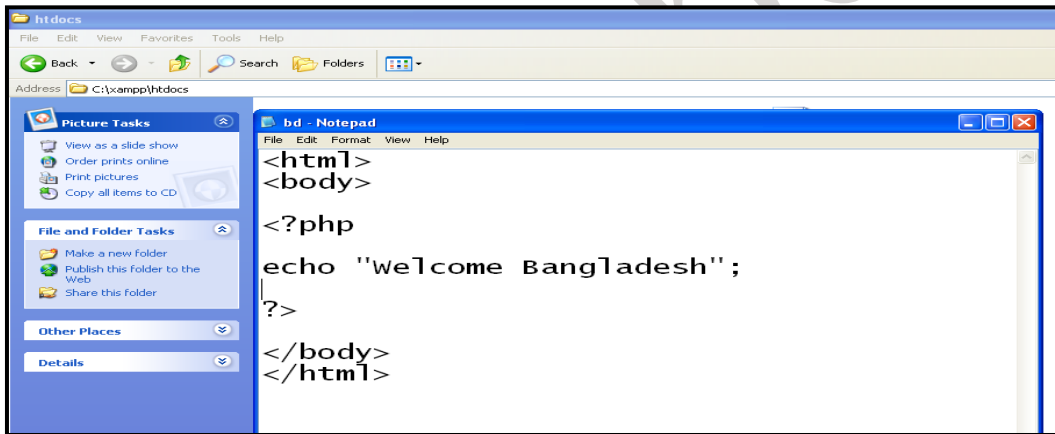
(চিত্রঃ ৪.৫.১ hello.php দিয়ে ফাইল এর নামকরণ)

অবজেক্ট অরিয়েটেড পিএইচপি

নিম্নের কোড গুলো hello.php ফাইল এ লিখুন। এই প্রোগ্রামের মাধ্যমে আমরা দেখবো, কিভাবে php এর মাধ্যমে কিছু লেখা বা টেক্সট আউটপুট হিসেবে ব্রাউজারে দেখা যায়।

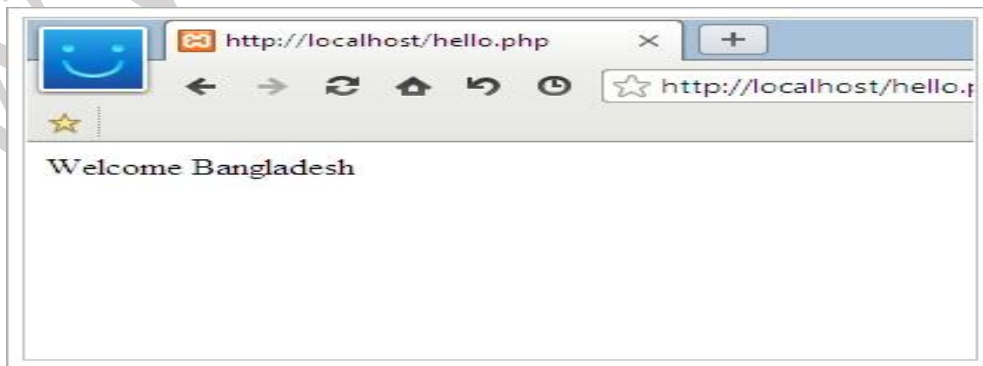
```
<html>
  <head>
</head>
  <body>
    <?php
      echo "Welcome Bangladesh";
    ?>
  </body>
</html>
```

প্রোগ্রাম নং ২ PHP এর মাধ্যমে কিছু লেখা বা টেক্সটকে আউটপুট হিসেবে দেখানোর জন্য উপরের প্রোগ্রামটিতে আমরা php স্ক্রিপ্টকে HTML এর Body Section এর ভেতর রেখেছি।



চিত্রঃ ৪.৫ প্রোগ্রাম ২ এর আউটপুট

এখন পূর্বেও মতোই আমরা এই ফাইলটি রান করবো। আবারো আপনি ব্রাউজারে চলে যান, সেখানে লিখুন <http://localhost/hello.php> এবং এন্টার দিন।



চিত্রঃ p4.5.2 hello.php

৪.৬ পিএইচপি এবং এইচটিএমএল

এখানে আমরা php স্ক্রিপ্টকে HTML এর Body Section এর ভেতর রেখেছি।

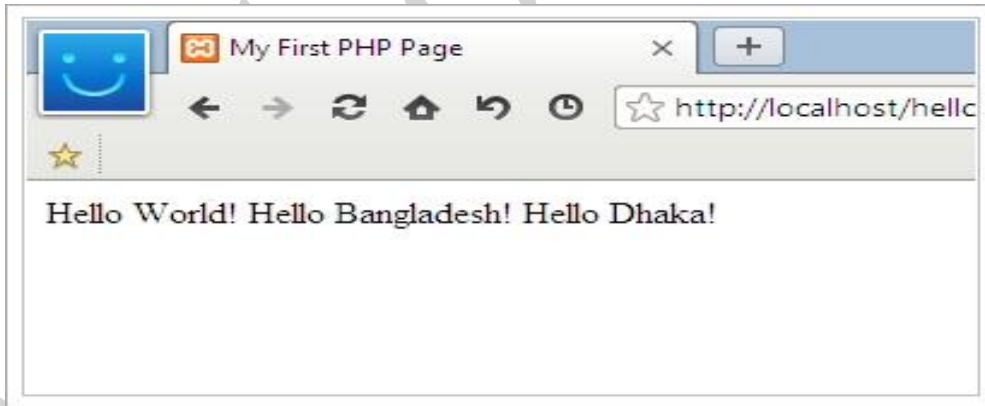
```
<html>
<head>
  <title>My First PHP Page</title>
</head>
<body>
  <?php
    echo "Hello World! ";
    echo "Hello Bangladesh! ";
    echo "Hello Dhaka! ";
  ?>
</body> </html>
```

প্রোগ্রাম নং ৩ HTML এর body ট্যাগ এর ভিতর <?php ?> ব্যবহার করে তার ভিতর PHP কোড লিখা

এখানে HTML এর body ট্যাগ এর ভিতর আমরা <?php ?> ব্যবহার করে তার ভিতর php কোড লিখেছি

```
echo "Hello World! ";
echo "Hello Bangladesh! ";
echo "Hello Dhaka! ";
```

এখন প্রোগ্রামটি রান করলে নিম্নের মতো আউটপুট প্রদর্শিত হবে।



চিত্রঃ ৪.৬ প্রোগ্রাম ৩ এর আউটপুট

উপরের প্রোগ্রামটি রান করলে এরকম আউটপুট আসবে।

তবে মনে রাখবেন, php তে কোন এক লাইন কোড অর্থাৎ statement লেখা শেষে সেমিকোলন (;) ব্যবহার করতে হবে আর ডাবল কোটেশন এর ভিতর যা থাকবে সেটি আউটপুট হিসেবে প্রদর্শিত হবে। যেমন উপরের কোডে লক্ষ্য করলে দেখবেন যে, ডাবল কোটেশন ("") এর ভিতর Hello Bangladesh! লিখা ছিল এবং সেটি আউটপুট হিসেবে দেখাচ্ছে।

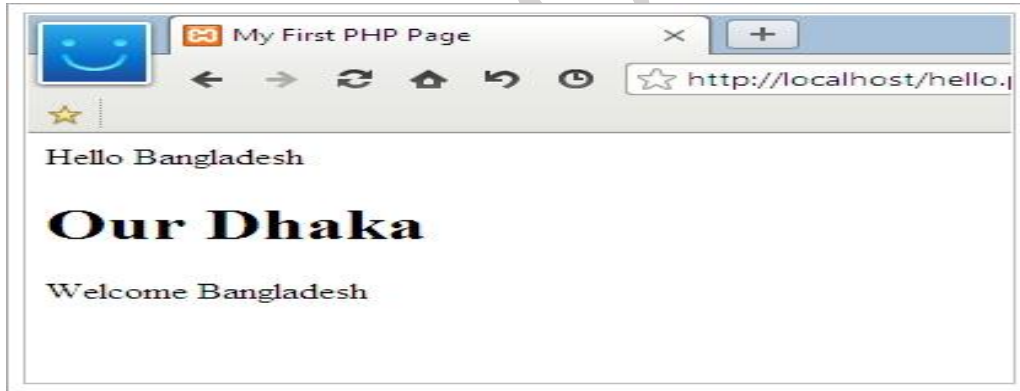
অবজেক্ট অরিয়েটেড পিএইচপি

এখন আপনাদেরকে আরেকটি প্রোগ্রাম দেখাবো। নিম্নের কোডগুলো আপনি একটি ফাইলে লিখে পূর্বের মতো সেভ করে রান করে দেখুন। নিম্নের প্রোগ্রামের মাধ্যমে PHP এবং HTML উভয় কোডের আউটপুট একসাথে দেখাবে।

প্রোগ্রাম নং ৪ PHP এবং HTML উভয় কোডের আউটপুট একসাথে দেখানোর জন্য প্রোগ্রাম

```
<html> <head>
  <title> My First PHP Page </title>
</head>
<body>
  <?php
    echo "Hello Bangladesh ";
  ?>
  <h1> Our Dhaka </h1>
  <?php
    echo "Welcome Bangladesh ";
  ?>
</body>
</html>
```

এই প্রোগ্রামটি রান করলে আউটপুটে দেখাবে



চিত্রঃ ৪.৬ প্রোগ্রাম ৪ এর আউটপুট

৪.৭ ক্লাস তৈরী

পিএইচপিতে ক্লাস তৈরীর জন্য Class কী-ওয়ার্ডটি ব্যবহার করতে হবে। ক্লাসের একটি নাম দিতে হয় সেটি আপনার পছন্দ অনুযায়ী দিতে পারেন। এরপর দু'টি দ্বিতীয় বন্ধনীর (Second Bracket) এর ভিতর ক্লাসের রিসোর্স (ক্লাসের কোড) গুলো থাকবে।

```
Class > কী-ওয়ার্ড nima > ক্লাসের নাম
{ শুরু সেকেন্ড ব্রাকেট
এখানে ক্লাসের প্রোপারটি এবং মেথড গুলো থাকবে।
```

} শেষের সেকেন্ড ব্রাকেট

উপরের nima নামক একটি ক্লাস তৈরী করে দেখানো হলো। আর আগেই বলেছি ক্লাস আমরা কেন ব্যবহার করবো। বুকবিডির ই-কমার্স বইটি পড়ুন নিজে নিজে ই-কমার্স ওয়েব সাইট তৈরী করুন।

```
<?php
    Class book
    {
    }
?>
```

উপরে বুক নামক একটি ক্লাস তৈরী করা হল।

৪.৮ অবজেক্ট তৈরী

অবজেক্টকে ক্লাসের instance ও বলা হয়। মনে রাখতে হবে ক্লাস ছাড়া অবজেক্ট তৈরী করা যায় না। অবজেক্ট তৈরী করার জন্য আপনি অবজেক্ট এর যে কোন একটি নাম দিবেন। অবশ্যই তার পূর্বে \$ (ডলার) সাইন দিতে হবে। যেমন \$d অর্থাৎ এখানে আমরা অবজেক্ট এর নাম দিলাম d আর এর পূর্বে \$ ডলার সাইন দিতে হবে।

```
$D =new nima;
```

\$D> অবজেক্টের নাম =new > এটি হচ্ছে কী-ওয়ার্ড nima > এটি ক্লাসের নাম ; > সেমিকোলন।

উপরে দেখানো হলো কী ভাবে nima ক্লাস এর একটি অবজেক্ট তৈরী করা হল। মনে রাখবেন একই ক্লাসের একের অধিক অবজেক্ট থাকতে পারে। এখন দেখাবো নিমা ক্লাসের আরেকটি অবজেক্ট এবং নাম হচ্ছে Sun

```
$sum = new nima;
```

তাহলে দেখলাম অবজেক্ট এর নামের পর ইকুয়াল (=) সাইন দিতে হবে। তারপর নিউ (new) কী-ওয়ার্ড লিখতে হবে এরপর স্পেস (Space) দিয়ে ক্লাসের নাম লিখতে হবে। অবশেষে সেমিকোলন (;) দিতে হবে।

৪.৯ অবজেক্ট এবং ক্লাস

```
<?php <==== php ট্যাগ শুরু করলাম
```

কী-ওয়ার্ড (এটি লিখতে হয়)



```
class bd <==== ক্লাস এর নাম bd
```

```
{ <==== দ্বিতীয় ব্রাকেট দিয়ে শুরু করতে হয়।
```

ভেরিয়েবল নেইম



```
public $city="dhaka"; <==== ভেরিয়েবল এর মান dhaka
```



মডিফায়ার (কী-ওয়ার্ড)

```
} <==== দ্বিতীয় ব্রাকেট দিয়ে শেষ (অর্থাৎ ক্লাসটি শেষ হল)
```

অবজেক্ট অরিয়েন্টেড পিএইচপি

ইকুয়াল সাইন

new এটি হচ্ছে কী-ওয়ার্ড
\$x=new bd; ← ক্লাস নেইম

এখানে ভেরিয়েবল x হচ্ছে একটি অবজেক্ট

অবজেক্ট x

echo \$x->city; ← bd ক্লাস এর ভেরিয়েবল

↑
নেইম স্পেস/অ্যারো অপারেটর

আউটপুট দেখানোর জন্য

?> ← পিএইচপি ট্যাগ শেষ করা হল।

8.10 অবজেক্ট অরিয়েন্টেড পিএইচপি প্রোগ্রাম

Class Mango

```
{  
    public $c=10;  
}  
$g=new mango;
```

কী-ওয়ার্ড → ভেরিয়েবল নেইম
ভেল্যু

এখন দেখবো কিভাবে g অবজেক্ট দিয়ে Mango ক্লাসের ভেরিয়েবলটি ব্যবহার করা যায়।

echo \$g->c;

এখন প্রোগ্রামটি রান (Execute) করলে আউটপুট হিসেবে ১০ প্রদর্শিত হবে।

প্রোগ্রাম: p5.php

```
<?php  
class mango  
{  
    public $c=10;
```

অবজেক্ট অরিয়েটেড পিএইচপি

```
}  
$g = new mango;  
echo $g->c;  
?>
```

উপরের প্রোগ্রামটির আউটপুট হিসেবে ১০ সংখ্যাটি দেখাবে। নিম্নে চিত্রের মাধ্যমে দেখানো হল।



চিত্রঃ ৪.১১ প্রোগ্রাম ৫ এর আউটপুট

প্রোগ্রাম: p6.php

```
<?php  
class orange{  
    public $value=100;  
    public $color="yellow";  
}  
$fruit= new orange;  
echo $fruit->value; // keep in mind dont need to put $ sign here before varibale  
echo $fruit->color; // dont use $ when access variable of class  
?>
```

উপরের প্রোগ্রামটি আউটপুট হবে 100 এবং Yellow;




চিত্রঃ ৪.১১ প্রোগ্রাম ৬ এর আউটপুট

ইন্টারনেটে ব্যবসা পরিচালনা...

ই-কমার্স
Your Future Business

Quick & Easy Way

More than just a book

With  রচনা ও সম্পাদনা ৪ বুকবিডি সিরিজ... www.bookbd.info

Download

অধ্যায়-৫ ভ্যালু, ভ্যারিয়েবল এবং কনস্ট্যান্ট

-
-
৫. ভ্যালু, ভ্যারিয়েবল, ভ্যারিয়েবল ডিক্লেয়ার, ডেটা টাইপস, কী-ওয়ার্ড, কনস্ট্যান্ট, কমেন্ট সম্পর্কে আলোচনা --
- ৫.১ ভূমিকা -----
- ৫.২ ভ্যালু -----
- ৫.৩ ভ্যারিয়েবল -----
- ৫.৪ পিএইচপিতে ভ্যারিয়েব ডিক্লেয়ার করার নিয়ম -----
- ৫.৫ পিএইচপি ভ্যারিয়েবল -----
- ৫.৬ ভ্যারিয়েবল (Variable) ভ্যালু (Value) এবং আউটপুট -----
- ৫.৭ পিএইচপিতে ডেটা টাইপস (Data Types) -----
- ৫.৮ ভ্যারিয়েবল এ ভ্যালিউ অ্যাসাইন -----
- ৫.৯ কী-ওয়ার্ড -----
- ৫.১০ কনস্ট্যান্ট (Constant) -----
- ৫.১১ কমেন্ট -----
- ৫.১২ সংক্ষিপ্ত অধ্যায় -----
- ৫.১৩ প্রশ্ন -----

৫.১ ভূমিকা :

প্রোগ্রামিং মূলত মান (ভ্যালু) নিয়ে কাজ করে আর এ ভ্যালু গুলো কোথাও না কোথাও রাখতে হয়। যেখানে ভ্যালুকে সংরক্ষণ করি তাকে আমরা ভ্যারিয়েবল বলে থাকি। আর এ মান গুলো বিভিন্ন ধরনের হয়ে থাকে, যেমন int, float, double, char, string, আর এগুলোকে ডেটা টাইপ বলা হয়। বাস্তবে এ ভ্যালু গুলো হচ্ছে এক ধরনের ডেটা (Data) এক ধরনের নির্দিষ্ট মান যেটা পরিবর্তন হওয়ার সম্ভাবনা নেই তাকে আমরা কনস্ট্যান্ট বা অপরিবর্তনীয় মান বলে থাকি।

৫.২ ভ্যালু

যেমন বলা হল একটি প্রোগ্রাম লিখ যেটি দুটো নাম্বারকে যোগ করবে। গুণ করবে অথবা চেক করবে। নাম্বার দুটির মধ্য কোন সংখ্যাটি ছোট ইত্যাদি যদি বলা হয়, আরেকটি প্রোগ্রাম লিখতে যেটি আউটপুট প্রদর্শিত করবে Bangladesh এখন প্রশ্ন হচ্ছে এই যে নাম্বারে দুটোর কথা বলা হল এ গুলো তাহলে কি? এ গুলো আমরা ডেটা/ভ্যালু বলতে পারি। তাহলে বলা যায় প্রোগ্রামিং ভ্যালু নিয়ে কাজ করে। নিম্নে কয়েকটি ভ্যালু লিখি ১০, ১২৫, ৬, Dhaka, p, ২৪.৫, ইত্যাদি

৫.৩ ভ্যারিয়েবল

ভ্যারিয়েবল ব্যবহার করা হয় মান (value) store করে রাখার জন্য। মান বলতে এখানে টেক্সট, নাম্বার, স্ট্রিং ইত্যাদি বুঝায়। ভ্যারিয়েবল (variable) হচ্ছে একটি কনটেইনার এর মতো, যেটি কোন কিছু কনটেইন অথবা ধারণ করে। আমরা এখানে কনটেইনার-যেটিকে ধারণ করে, সেটিকে value বলছি আর কনটেইনারকে বলছি ভ্যারিয়েবল। ধরুন, একটি পাত্রে কিছু আপেল রয়েছে, এখানে পাত্রটি হচ্ছে ভ্যারিয়েবল আর পাত্রের মধ্যে যে আপেল গুলো রয়েছে সেগুলো হচ্ছে মান (Value)। অর্থাৎ আপেলকে এখানে বলা যায়, পাত্রের মান অর্থাৎ ভ্যারিয়েবল এর মান। একটি প্রোগ্রামে ভ্যারিয়েবল বার বার ব্যবহার করা যায় (অর্থাৎ reuse করা যায়)। ভ্যারিয়েবল প্রোগ্রামে ব্যবহার করা হয় কোন একটি ভ্যালিউকে Represent করার জন্য। প্রোগ্রামে ভ্যারিয়েবল ব্যবহার এর কারণে কোন মানকে বার বার লিখার প্রয়োজন হয় না। এতে করে প্রোগ্রামের সাইজ কমে আসে। কোন একটি প্রোগ্রামের যখন, যেখানে প্রয়োজন সেখানে Variable কে কল (call) করা হয়। ভ্যারিয়েবল ডিক্লেয়ার করার জন্য প্রত্যেকটি ল্যাংগুয়েজের নিজস্ব রুলস রয়েছে। যেমন সি, সি++, জাভা, ডটনেট এবং পিএইচপি প্রোগ্রামিং ল্যাংগুয়েজ। প্রত্যেক ল্যাংগুয়েজের ভ্যারিয়েবল Declare করার নিয়ম ভিন্ন, তবে আসল Concept এবং কাজ একই। নিম্নে কয়েকটি ভ্যারিয়েবল এর উদাহরণ দেওয়া হল:

```
$Variable_Name=Value;
```



ভ্যারিয়েবল এর নাম



ভ্যারিয়েবল এর মান

এই ভ্যারিয়েবল এর নাম আপনি যেকোন কিছু লিখতে পারেন এবং সেটিতে যে কোন মান (value) ও রাখতে পারেন।

```
$bd=5;
```

এখানে bd হচ্ছে ভ্যারিয়েবল এবং 5 হচ্ছে উক্ত bd ভ্যারিয়েবল এর মান। মনে রাখবেন, ভ্যারিয়েবল লেখার পূর্বে অবশ্যই ডলার সাইন (\$) ব্যবহার করতে হবে।

নিচের প্রোগ্রামে txt এবং number নামের দুটি ভ্যারিয়েবল ডিক্লেয়ার করা হয়েছে এবং এদের (value) হিসেবে যথাক্রমে Bangladesh এবং 16 দেওয়া হয়েছে। এখন এই ভ্যারিয়েবল দুটিকে কল করার জন্য আমরা echo

অবজেক্ট অরিয়েটেড পিএইচপি

\$txt; এবং echo \$number; লিখেছি। এবং আউটপুট হিসেবে উক্ত ভ্যারিয়েবল দুটির মান যথাক্রমে Bangladesh16 দেখতে পাবো।

প্রোগ্রাম নং ৭ (p7.php) ভ্যারিয়েবল ডিক্লেয়ারের মাধ্যমে ভ্যালিউ অ্যাসাইন করা ও আউটপুট দেখানো

```
<?php
```

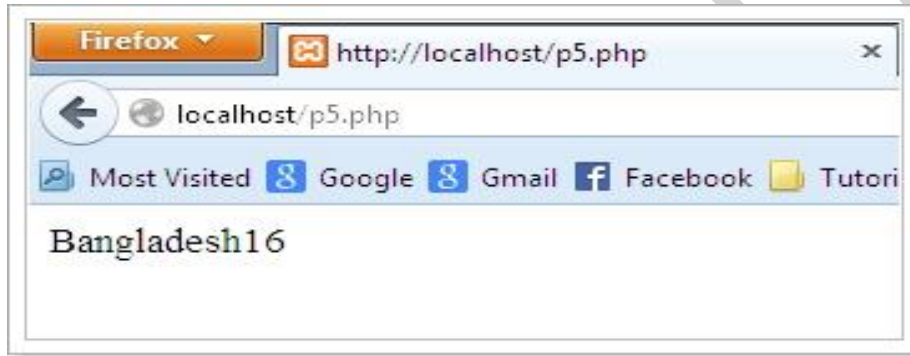
```
    $txt = "Bangladesh";
```

```
    $number = 16;
```

```
    echo $txt;
```

```
    echo $number;
```

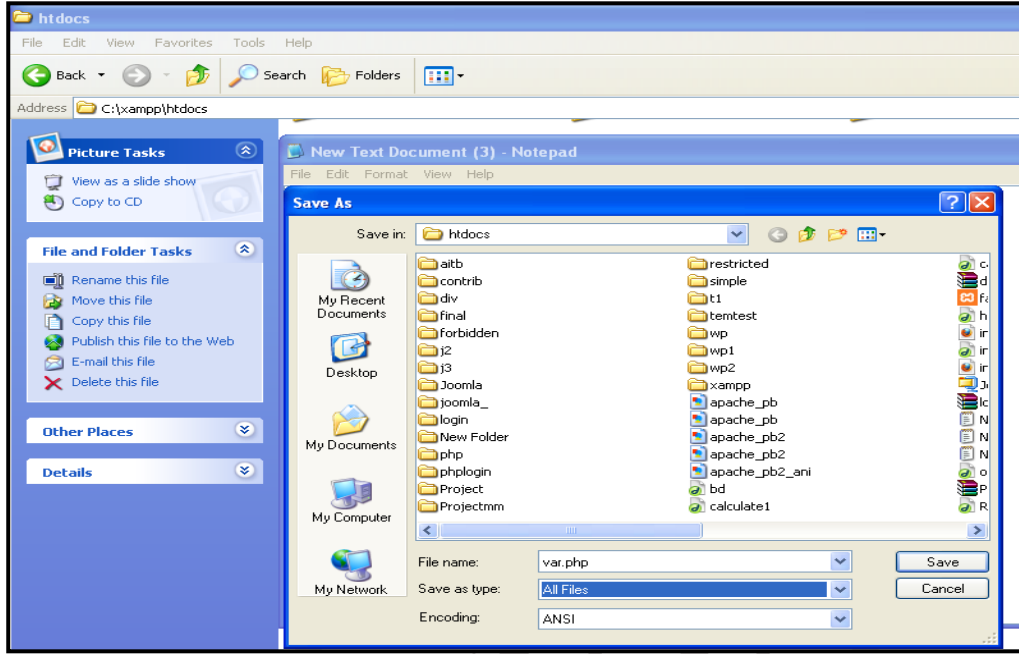
```
?>
```



চিত্রঃ ৫.৩ প্রোগ্রাম ৭ এর আউটপুট

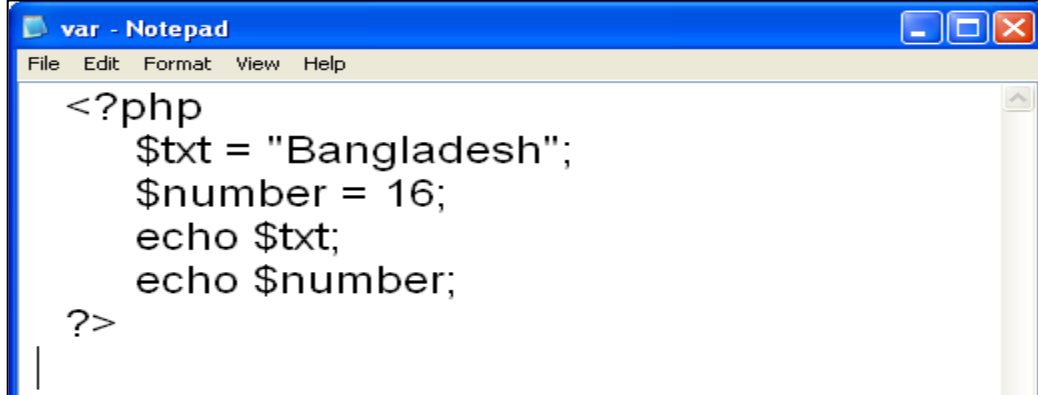
একটি নোটপ্যাড ওপেন করে উপরের কোড গুলো লিখুন এবং সেটিকে যেকোন নাম দিয়ে ডট (.php) extension সহ সেভ করুন। যেটি পূর্বে কয়েকবার দেখানে হয়েছে। তারপরও আরও একবার দেখানো হলো। আপনি কম্পিউটারের C ড্রাইভে XAMPP ফোল্ডারটি খুঁজে বের করুন এবং সেখান থেকে htdocs ফোল্ডারের ভিতর চলে যান। মনে রাখবেন, আপনি যদি XAMPP সফটওয়্যারটি অন্য কোন ড্রাইভে ইনস্টল করে থাকেন, তাহলে কিন্তু C ড্রাইভে গিয়ে পাবেন না। বরং, যে ড্রাইভে ইনস্টল করেছেন সেখানে যেতে হবে। এখন htdocs ফোল্ডারের ভিতর মাউসের রাইট বাটন ক্লিক করে একটি Text Document (টেক্সট ডকুমেন্ট) ফাইল ওপেন করুন এবং ফাইলটিতে ডাবল ক্লিক করে ফাইলটির ফাইল অপশনে গিয়ে Save as (সেভ এজ) এ ক্লিক করে এটিকে var.php নামে সেভ করুন এবং Save as type থেকে অবশ্যই All Files সিলেক্ট করে দিবেন।

অবজেক্ট অরিয়েটেড পিএইচপি



চিত্র ৫.৩.১ : var.php ফাইল

এখন ফাইলে উপরের কোডগুলো (p5.php) টাইপ করুন।



চিত্র ৫.৩.২ : var.php ফাইলে php কোড

এখন ব্রাউজারে গিয়ে localhost/var.php লিখে এন্টার দিন, আউটপুট হিসেবে Bangladesh16 দেখতে পাবেন।

নিম্নে একটি প্রোগ্রাম লিখা হচ্ছে যেখানে দেখানো হয়েছে কিভাবে ক্লাসের সব গুলো ভ্যারিয়েবলকে এক সাথে প্রিন্ট করা যায়।

প্রোগ্রাম : p8.php

```
<?php
class p{
    public $d=10;
```

```
public $e="Bookbd for Bangla Book";  
}  
$g = new p;  
var_dump($g);  
echo $g->d;  
echo $g->e;  
?>
```

var_dump() ফাংশনটির ভিতরে অবজেক্ট লিখলে। যে ক্লাসের অবজেক্ট সে ক্লাসের সকল ভ্যারিয়েবল প্রিন্ট হয়ে যাবে। যেমন var_dump (\$g) এখানে p ক্লাসের ভ্যারিয়েবল d এবং e এর মান এবং ডেটা টাইপ সহ আউটপুট প্রদর্শিত হবে। নিম্নে চিত্রের মাধ্যমে দেখানো হল।



চিত্রঃ ৫.৩ প্রোগ্রাম ৮ এর আউটপুট

৫.৪ পিএইচপিতে ভ্যারিয়েব ডিক্লেয়ার করার নিয়ম

- PHP ভ্যারিয়েবল অবশ্যই letter অথবা underscore (“_”) দিয়ে শুরু করতে হবে।
- ভ্যারিয়েবল নেইম শুধুমাত্র আলফা নিউমেরিক (Alpha numeric) ক্যারেক্টারস এবং Underscore (a-z, A-Z, 0-9, _) কনটেইন (Contain) করতে পারবে।
- ভ্যারিয়েবল এর নামে space ব্যবহার করা যাবে না। যদি ভ্যারিয়েবল এর নাম একের অধিক শব্দের হয়, তখন underscore (“_”) অথবা হাইফেন (“-”) ব্যবহার করে সেগুলোকে আলাদা করা যেতে পারে।
- ভ্যারিয়েবল এর নামে Capitalization ও ব্যবহার করা যাবে।
- ভ্যারিয়েবল এর নামের শুরুতে অবশ্যই dollar (\$) সাইন থাকতে হবে।
- ভ্যারিয়েবল এর নামের প্রথম অক্ষরটি নাম্বার হতে পারবে না, সেটি অবশ্যই Letter অথবা Underscore দিয়ে শুরু হতে হবে।

PHP ভ্যারিয়েবল Case-sensitive অর্থাৎ php syntax মেনে variable declare না করলে এরর (error) দেখাবে, আপনার প্রোগ্রাম রান করবে না।

৫.৫ পিএইচপি ভ্যারিয়েবল

এখন আমরা php ভ্যারিয়েবল এর নিয়মগুলো মেনে ভ্যারিয়েবল তৈরি (Declare) করবো : \$txt

এটি শুরু হয়েছে \$ সাইন দিয়ে এবং প্রথম Letter হচ্ছে (a-z) এর মধ্যে। সুতরাং, এটি একটি ভ্যালিড ভ্যারিয়েবল। \$Txt

এটি \$ সাইন দিয়ে শুরু হয়েছে এবং প্রথম অক্ষর A থেকে Z এর মধ্যে। এখানে ভ্যারিয়েবল Capital letter দিয়ে শুরু হয়েছে। সুতরাং এটিও একটি ভ্যালিড ভ্যারিয়েবল। \$_txt

এটিও \$ সাইন দিয়ে শুরু হয়েছে এবং ভ্যারিয়েবল এর নামের প্রথমে Underscore আছে, সুতরাং এটিও একটি ভ্যালিড ভ্যারিয়েবল। \$5txt

এটি \$ সাইন দিয়ে শুরু হয়েছে ঠিকই, তবে এটি ভ্যালিড ভ্যারিয়েবল না। কারণ, এর প্রথম লেটার হচ্ছে নাম্বার আর শর্ত হচ্ছে প্রথম লেটার নাম্বার হওয়া যাবে না। \$bd5

এটি একটি ভ্যালিড ভ্যারিয়েবল। কারণ, এখানে নাম্বার 5 ব্যবহার করা হয়েছে ঠিকই, তবে তা ভ্যারিয়েবল এর নামের শুরুতে না। অর্থাৎ, নাম্বার-ভ্যারিয়েবল এর মাঝখানে এবং শেষে থাকতে পারে।

নিম্নে কিছু ভ্যালিড ভ্যারিয়েবল লেখা হল :

\$V

\$car

\$dhaka

\$position

\$num

\$num5

\$num56

\$num5bd

\$_num

\$num_

\$num_56

\$dhaka_bd

\$March_Salary

\$March12

\$salaryEmployee

\$add_two_number

\$addTwoNumber

\$addtwonumber

\$employee_name

\$studentName

\$studentaddress

\$Student

\$a

\$A

\$aa

\$ab

\$b

\$b_a

\$a6

\$xyz

\$x
\$aR
\$x100
\$m12265

আমরা এখন উপরের মতো করে ভ্যারিয়েবল ডিক্লেয়ার করতে পারবো। শুধু নিয়ম গুলো মেনে যে কোন নামে variable ডিফাইন করা যায়।

\$num=56; ← ভ্যারিয়েবল লিখা শেষে অবশ্যই সেমিকোলন দিতে হবে।



এই Equal(=) সাইন এর মানে হচ্ছে, 56 value টি num ভ্যারিয়েবল এ assign করে দেওয়া হল।

\$name= "Rony ";

Rony নামটি name ভ্যারিয়েবল এ রাখা হল। এই value টি একটি স্ট্রিং, যার জন্য এটিকে ডাবল কোটেশন (" ") এর ভিতর লেখা হয়েছে।

\$sign= 'a ';

এখানে a হচ্ছে একটি character, যার জন্য এটিকে single quotation (' ') এর ভিতর রাখা হয়েছে।

\$nm=20;

এখানে nm ভ্যারিয়েবল এ value হিসেবে 20 রাখা হয়েছে এবং এটি একটি number হওয়ায় কোন কোটেশন (quotation) ব্যবহার করা হয়নি।

এখন আমরা আউটপুট হিসেবে ব্রাউজারে উক্ত ভ্যারিয়েবল গুলোতে অ্যাসাইন করা value দেখতে পাব, ভ্যারিয়েবল এর নাম না।

এখন আমরা দেখবো যেগুলো ভ্যারিয়েবল না :

\$5n

এখানে ভ্যারিয়েবল টি \$ সাইন দিয়ে শুরু হয়েছে ঠিক আছে, কিন্তু প্রথম লেটার নাম্বার হওয়াতে এটি ভ্যারিয়েবল না।

\$employee salary

এখানে স্পেস (space) ব্যবহার করা হয়েছে, এজন্য এটি ভ্যারিয়েবল না।

\$emp\$

ডলার সাইন শুধুমাত্র ভ্যারিয়েবল এর শুরুতে ব্যবহার করা যাবে, পরে কোথাও না। এজন্য এটি ভ্যারিয়েবল না।

\$emp@n

এখানে @ সাইন ব্যবহার করা হয়েছে, যেটি ভ্যারিয়েবলে ব্যবহার করা যাবে না। এজন্য এটিও PHP ভ্যারিয়েবল না।

ভ্যারিয়েবল না এমন কিছু উদাহরণ :

\$56a
\$--ab
\$ab@
\$@ab
\$name!
\$ name
var
var\$
bd#
*bd

```
$student*name  
$bd-dhaka (This is hyphen)  
$bd%  
$dhaka\
```

খুব সহজভাবে ওয়েব সাইট ডিজাইন শিখতে পড়ুন বুকবিডি সিরিজের এইচটিএমএল এবং সিএসএস বই দুটো।

৫.৬ ভ্যারিয়েবল (Variable) ভ্যালু (Value) এবং আউটপুট

এখন আমরা দেখবো, কিভাবে variable এ value রাখতে হয় এবং তা আউটপুটে দেখাতে হয়।

\$bd = 576; ← মিকোলন দিয়ে শেষ করতে হয়।

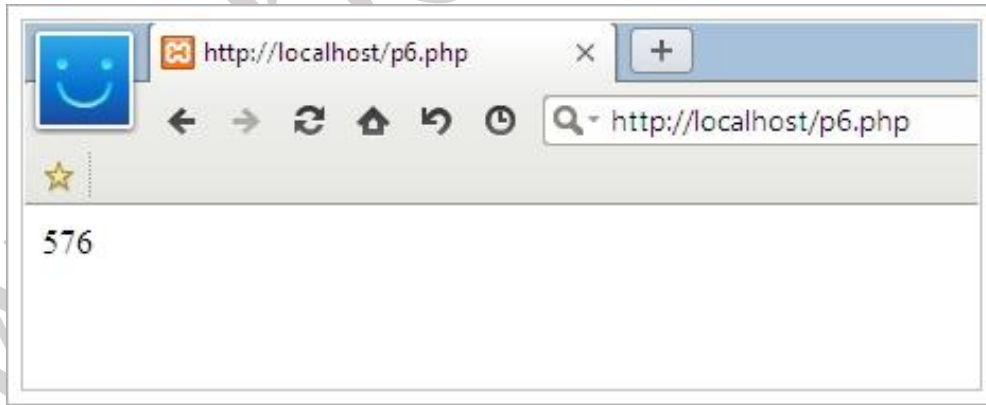
↑ ↑ ↑
ভ্যারিয়েবল ইকুয়াল (=) সাইন ভ্যালিউ

তাহলে বলা যায়, ইকুয়াল সাইন এর পূর্বে (বামে) যেটি থাকবে সেটি হচ্ছে variable এবং পরে (ডানে) যেটি থাকবে সেটি হচ্ছে value। value সব সময় ভ্যারিয়েবল এ Assign (চলে আসে) হয়।

এখন আমি যদি আউটপুটে 576 দেখতে চাই, তাহলে echo 576; লিখলে তা দেখাবে।

প্রোগ্রাম নং ৯ (p9.php) আউটপুট হিসেবে 576 দেখার জন্য

```
<?php  
echo 576;  
?>
```



চিত্রঃ ৫.৬ প্রোগ্রাম ৯ এর আউটপুট

উপরের প্রোগ্রামটিতে লক্ষ্য করুন, আপনি ইকো (echo) এর পরে যা লিখবেন (576 সংখ্যা হওয়ায় কোডেশন ব্যবহার করা হয়নি) তাই শো করবে। কিন্তু আমি এখন চাচ্ছি, value টিকে সরাসরি ইকোতে না লিখে কোন একটি ভ্যারিয়েবল এর মাধ্যমে কল করতে। এজন্য প্রথমে value টিকে কোন ভ্যারিয়েবল এ রাখতে (Assign) হবে। চলুন, এখন 576 ভ্যালিউটিকে যে কোন একটি ভ্যারিয়েবল এ রাখি।

```
$num= 576;
```

ভ্যারিয়েবল এবং সবশেষে একটি সেমিকোলন থাকবে।

```
echo $num;
```

↑ ↑ ↑
Space থাকবে

এটি আউটপুটের জন্য লিখতে হয়

চিত্র : (৪.৪.১) ভ্যারিয়েবল প্রিন্ট করা।

নিচের প্রোগ্রামে এ \$num ভ্যারিয়েবলটি ডিক্লেয়ার করা হয়েছে এবং এর মান দেওয়া হয়েছে 576। এখানে echo \$num লেখা হয়েছে, এবং যেহেতু \$num-এর মান (value) হচ্ছে 576 তাই আউটপুটে 576 দেখাবে।

প্রোগ্রাম নং ১০ (p10.php) Value সহ ভ্যারিয়েবল ডিক্লেয়ার করে ভ্যারিয়েবলের মান আউটপুট হিসেবে দেখানো

```
<?php  
class pp{  
    public $num=576;  
}  
$ob = new pp;  
echo $ob-> num;  
?>
```

উপরের ১০ নং প্রোগ্রামের Output: 576

উপরের প্রোগ্রামটি রান করলে দেখবেন আউটপুটে \$num দেখাচ্ছে না, দেখাবে এর মান, অর্থাৎ 576।

নিচের প্রোগ্রামে name নামের একটি ভ্যারিয়েবল ডিক্লেয়ার করা হয়েছে, এবং এর মান হিসেবে একটি string অর্থাৎ jasika দেওয়া হয়েছে, যেহেতু এখানে echo \$name লিখা হয়েছে, এবং যেহেতু \$name এর মান jasika তাই আউটপুট হিসেবে jasika দেখাবে।

প্রোগ্রাম নং ১১ (p11.php) ভ্যারিয়েবলের ভ্যালিউ হিসেবে স্ট্রিং ব্যবহার করা

```
<?php  
$name="jasika";  
echo $name;  
?>
```



চিত্রঃ ৫.৬ প্রোগ্রাম ১১ এর আউটপুট

ভ্যালিউকে ডাবল কোটেশন দিয়ে লিখা মানে হচ্ছে, এটি একটি স্ট্রিং যা পরবর্তীতে আলোচনা করা হবে।
নিচের প্রোগ্রামের মাধ্যমে কিভাবে একটি ক্যারেক্টারকে আউটপুট হিসেবে ডিসপ্লে করা যায়, তা দেখানো হল।

প্রোগ্রাম নং ১২ (p12.php) ভ্যারিয়েবলের ভ্যালিউ হিসেবে ক্যারেক্টার ব্যবহার করা

```
<?php
class pp{
    public $n='b';
}
$obj = new pp;
// var_dump($g);
echo $obj->n;
?>
```

উপরের ১২ নং প্রোগ্রামের : Output: b

এখানে \$n হচ্ছে একটি ভ্যারিয়েবল এবং 'b' (মান একটি মাত্র ক্যারেক্টার হওয়ায় সিঙ্গেল কোটেশন এর মধ্যে লিখা হয়েছে।) হচ্ছে একটি ক্যারেক্টার। উক্ত প্রোগ্রামে আউটপুট হিসেবে ডিসপ্লে হবে b।

নিচের প্রোগ্রামে name নামের একটি ভ্যারিয়েবল ডিক্লেয়ার করা হয়েছে, এবং এর মান হিসেবে একটি string অর্থাৎ ronay ahmed দেওয়া হয়েছে, যেহেতু এখানে echo \$name লেখা হয়েছে, এবং যেহেতু \$name এর মান ronay ahmed তাই আউটপুট হিসেবে ronay ahmed দেখাবে।

```
<?php
$name="ronay ahmed";
echo $name;
?>
```

উপরের প্রোগ্রামের : Output: ronay ahmed

নিচের প্রোগ্রামে আলাদা ভাবে দুইটি ভ্যারিয়েবল ডিক্লেয়ার করে তাদের মান অ্যাসাইন করা হয়েছে এবং ভ্যালিউ আউটপুট করার জন্য echo ব্যবহার করে উক্ত ভ্যারিয়েবল দুটিকে কল করা হয়েছে।

প্রোগ্রাম নং ১৩ (p13.php)

```
<?php
$name="ronay ahmed";
echo $name;
$m="South Asian ICT";
echo $m;
?>
```


অবজেক্ট অরিয়েটেড পিএইচপি



চিত্রঃ ৫.৬ প্রোগ্রাম ১৩ এর আউটপুট

প্রোগ্রাম নং ১৪ (p14.php) দুইটি ভ্যারিয়েবল আলাদাভাবে ডিক্লেয়ার করে ভ্যারিয়েবল গুলোর Value আউটপুট হিসেবে দেখানো

```
<?php
$name="ronay ahmed";
    echo $name;
$m="South Asian ICT";
    echo $m;
$mmm="South Asian ICT Gulshan Dhaka";
    echo $mmm;
?>
```



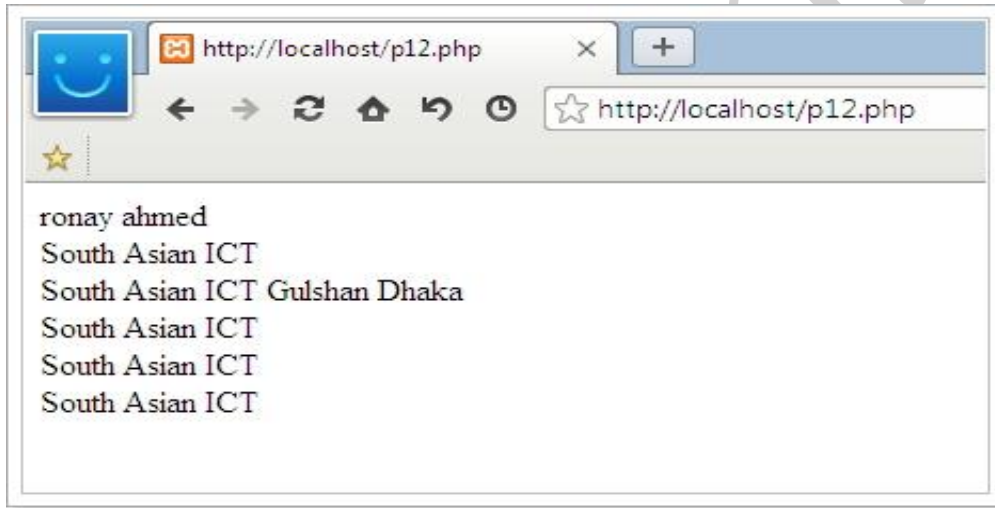
চিত্রঃ ৫.৬ প্রোগ্রাম ১৪ এর আউটপুট

প্রোগ্রাম নং ১৫ (p15.php) তিনটি ভ্যারিয়েবল আলাদাভাবে ডিক্লেয়ার করে ভ্যারিয়েবল গুলোর Value আউটপুট হিসেবে দেখানো

অবজেক্ট অরিয়েটেড পিএইচপি

নিচের প্রোগ্রামের মাধ্যমে একই ভেরিয়েবলের মানকে একাধিক বার আউটপুট হিসেবে দেখানো হচ্ছে।

```
<?php
    $name="ronay ahmed";
        echo $name;
    $m="South Asian ICT";
        echo $m;
    $mm="South Asian ICT Gulshan Dhaka";
        echo $mm;
        echo $m;
        echo $m;
        echo $m;
?>
```



চিত্রঃ ৫.৬ প্রোগ্রাম ১৫ এর আউটপুট

প্রোগ্রাম নং ১৬ (p16.php) একই ভ্যারিয়েবলের মানকে একাধিক বার আউটপুট হিসেবে দেখানো

নিচের প্রোগ্রামে num নামের একটি ভ্যারিয়েবল ডিক্লেয়ার করা হয়েছে এবং এর ভ্যালিউ হিসেবে 45 দেওয়া হয়েছে। যেহেতু echo num লেখা হয়েছে, এবং যেহেতু num এর মান 14 তাই আউটপুট হিসেবে 45 দেখাবে।

```
<?php
$num=45;
echo $num;
?>
```

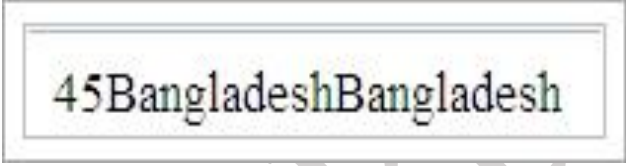
উপরের ১৬ নং প্রোগ্রামের : Output: 45

অবজেক্ট অরিয়েটেড পিএইচপি

নিচের প্রোগ্রামগুলোতে (প্রোগ্রাম নং 15 থেকে প্রোগ্রাম নং 20) ভ্যারিয়েবলের মান আউটপুটে দেখানোর জন্য echo এর মাধ্যমে কল করা হচ্ছে, আবার ভ্যারিয়েবল ডিক্লেয়ার না করে echo মাধ্যমে সরাসরি আউটপুট দেখানো হচ্ছে।

প্রোগ্রাম নং ১৭ (p17.php) ভ্যারিয়েবলের ভ্যালিউ 45 দিয়ে তার আউটপুট দেখানো

```
<?php
    $num=45;
    echo $num;
    echo "Bangladesh";
    $country="Bangladesh";
    echo $country;
?>
```




45BangladeshBangladesh

চিত্রঃ ৫.৬ প্রোগ্রাম ১৭ এর আউটপুট

প্রোগ্রাম নং ১৮ (p18.php)

```
<?php
    $num=45;
    echo $num;
    echo "Bangladesh";
    $country="Bangladesh";
    echo $country;
    $add=456123;
    echo $add;
?>
```

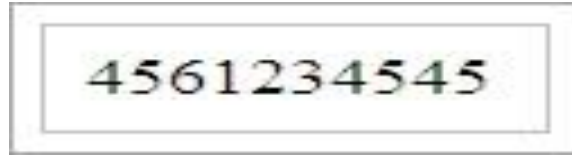


45BangladeshBangladesh456123

চিত্রঃ ৫.৬ প্রোগ্রাম ১৮ এর আউটপুট

প্রোগ্রাম নং ১৯ (p19.php)

```
<?php
$num=45;
$add=456123;
echo $add;
echo $num;
$num=45;
echo $num;
?>
```



চিত্রঃ ৫.৬ প্রোগ্রাম ১৯ এর আউটপুট

প্রোগ্রাম নং ২০ (p20.php)

```
<?php
$integer_var = 3;
echo $integer_var;
?>
```

উপরের ২০ নং প্রোগ্রামের : **Output: 3**

প্রোগ্রাম নং ২১ (p21.php)

```
<?php
$integer_var = 3;
$str_var = "apples";
echo $integer_var." ".$str_var;
?>
```



চিত্রঃ ৫.৬ প্রোগ্রাম ২১ এর আউটপুট

```
<?php
$city = "New York";
```

```
$var_container = "city"; //$var_container will store the variable $city  
echo "CONTAINER's var: " . $var_container;  
echo "<br />";  
echo "CONTAINER's value: " . $$var_container;  
echo "<br />";  
echo "VAR city: " . $city;
```

?>



৫.৭ পিএইচপিতে ডেটা টাইপস (Data Types)

প্রোগ্রামিং এ বিভিন্ন ধরনের মান (ডেটা) থাকার কারণে এ ধরনের টাইপ ও ভিন্ন হয়। যেমন ১০ এবং ২৬.৫ এখানে ১০ এক ধরনের ডেটা আবার ২৬.৫ হচ্ছে অন্য ধরনের ডেটা। আবার ২।৬ এবং a এর মধ্যে, ৬ এক ধরনের অপর দিকে a হচ্ছে অন্য ধরনের ডেটা। এ সমস্ত ডেটার উপর ভিত্তি করে ডেটার টাইপ ও ভিন্ন হয়।

বেশির ভাগ প্রোগ্রামিং এ ডেটা টাইপ লিখে দিতে হয়। কিন্তু PHP প্রোগ্রামিং এ কোন ডেটা টাইপ লিখতে হয় না। যেমন সি প্রোগ্রামিং এ যদি কোন number লিখি, তাহলে লিখতে হয়

```
int a= 7;  
↑      ↑      ↑  
ডেটা টাইপ  ভ্যারিয়েবল  ভ্যালিউ
```

কিন্তু PHP তে তা লাগে না যেমন \$a=7;

এখানে কোন ডেটা টাইপ নেই।

PHP তে একটি ভ্যারিয়েবল এ যে কোন ধরনের ভ্যালিউ রাখা যায়, কিন্তু অন্যান্য ল্যাংগুয়েজে সেটা যায় না।

PHP তে ডেটা টাইপটা অটোমেটিক্যালি ভ্যারিয়েবল এ set হয়, অর্থাৎ আপনি যে টাইপের ডেটা (মান) ভ্যারিয়েবল এ রাখবেন, ভ্যারিয়েবল সেই টাইপের হয়ে যাবে।

যেমন:

\$a= 75; অটোমেটিক্যালি ইন্টিজার (integer) টাইপ হয়ে যাবে।
\$bd=75.6; এটি অটোমেটিক্যালি ফ্লোট (float) টাইপ হয়ে যাবে।
\$n= "Mijan"; এটি স্ট্রিং (string) টাইপ হয়ে যাবে।
\$ch= 'p'; এটি ক্যারেক্টার (character) টাইপ হয়ে যাবে।
নিচের প্রোগ্রামে বিভিন্ন ডেটা টাইপ ব্যবহার করা হয়েছে।

```
<?php
    $myAge = 16;           // a PHP Integer - always available
    $yourAge = 15.5;      // a PHP Float - always available
    $hasHair = true;      // a PHP Boolean - always available
    $greeting = "Hello World!"; // a PHP String - always available
?>
```

প্রোগ্রাম নং ২২ (p22.php) বিভিন্ন ডেটা টাইপ ব্যবহার করে প্রোগ্রাম লেখা

এখন দেখবো কিভাবে ক্লাসের ভিতর ভ্যালু প্রিন্ট হয়। সে জন্য আমাদেরকে একটি ফাংশন নিতে হবে। এখানে

```
<?php
class m{
    public function demo() // as rules write public than wirte function keyword
    and after that function name {
        echo "BookBd";
    }
}
$obj = new m;
$obj->demo();
?>
```



চিত্রঃ ৫.৭ প্রোগ্রাম ২২ এর আউটপুট

৫.৮ ভ্যারিয়েবল এ Value অ্যাসাইন

সুবিধার জন্য Valueকে ভ্যারিয়েবল এ স্টোর করে রাখা হয়। অ্যাসাইনমেন্ট অপারেটর এর সাহায্যে ভ্যারিয়েবলে ভ্যালিউকে স্টোর করা হয়।

[Assign Operator " = "] "= " হচ্ছে অ্যাসাইনমেন্ট অপারেটর।



Variable_Name=constant;



এটি যে কোন নাম হতে পারে।

এটি যে কোন মান বা Value হতে পারে।

```
$product= "apple";
```

এখানে product ভ্যারিয়েবলে apple কে Value হিসেবে অ্যাসাইন করা হয়েছে।

```
$price= 70.5;
```

price ভ্যারিয়েবলে 70.5 দামটি রাখা হয়েছে।

একই ভ্যারিয়েবল নামে একাধিক Value রাখা যায়, তবে শেষ Value টি সব সময় পাওয়া যাবে।

```
$a= 5;
```

```
$a=6;
```

```
$a=10;
```

এখন যদি আপনি জানতে চান, a এর মান কত? তা হলে উত্তর হবে 10. কারণ একই ভ্যারিয়েবল নামে একাধিক মান রাখলে শেষ মানটাই পাওয়া যাবে।

নিচের প্রোগ্রামে a ভ্যারিয়েবলের Value হিসেবে hello অ্যাসাইন করা হয়েছে।

```
<?php
  Public class dk{
    public $a = "hello";
  }
?>
```

নিচের প্রোগ্রামে p এবং number নামের দুইটি ভ্যারিয়েবল ডিক্লেয়ার করে এদের Value অ্যাসাইন করা হয়েছে, কিন্তু যেহেতু এখানে শুধুমাত্র echo \$number লেখা হয়েছে-তাই এখানে শুধুমাত্র number ভ্যারিয়েবলের Value আউটপুট হিসেবে দেখাবে।

প্রোগ্রাম নং ২৩ (p23.php) ভ্যারিয়েবলে ভ্যালিউ অ্যাসাইন করা

```
<?php
Public class dom{
Public function k() {
  $p=10;
  $number=12;
  echo $number;
}}
P=new dom;
```

```
p->k();  
?>
```

এখন আমরা দেখবো, একটি ভ্যারিয়েবল এর মান কিভাবে অন্য ভ্যারিয়েবল এ রাখতে অথবা অ্যাসাইন করতে হয়।

\$p=10; এখানে p এর মান হচ্ছে 10.

\$number=\$p; এখানে number ভ্যারিয়েবল এর মান কিন্তু \$p নয়, এর মান হচ্ছে 10. অর্থাৎ \$p ভ্যারিয়েবল এর Value টি number ভ্যারিয়েবলে Assign হবে।

প্রোগ্রাম নং ২৪ (p24.php) একাধিক ভ্যারিয়েবলের Value ডিক্লেয়ার করে শুধুমাত্র একটি ভ্যারিয়েবলের Value আউটপুট হিসেবে দেখানো

```
<?php  
$p=10;  
$number=$p;  
echo $number;  
?>
```

উপরের ২৪ নং প্রোগ্রামটির **Output: 10**

প্রোগ্রাম নং ২৫ (p25.php) একটি ভ্যারিয়েবলের মান অন্য ভ্যারিয়েবলে রাখা অথবা অ্যাসাইন করা

```
<?php  
$p=10;  
$number=$p;  
echo $number;  
$p=$number;  
echo $p;  
$p=542;  
echo $p;  
?>
```

উপরের ২৫ নং প্রোগ্রামটির **Output: 10 10 542**

প্রোগ্রাম নং ২৬ (p26.php) একটি ভ্যারিয়েবলের মান অন্য ভ্যারিয়েবলে রাখা অথবা অ্যাসাইন করা

```
<?php  
$p=10;  
$number=$p;  
echo $number;  
$p=$number;
```


অবজেক্ট অরিয়েটেড পিএইচপি

```
echo $p;  
$p=542;  
echo $p;  
$number=$p;  
echo $number;  
?>
```

উপরের ২৬ নং প্রোগ্রামটির : **Output:** 10 10 542 542

৫.৯ কী-ওয়ার্ড

কম্পিউটার প্রোগ্রামিং এ কী-ওয়ার্ড হচ্ছে এক ধরনের word অথবা identifier যেটি প্রোগ্রামিং ল্যাংগুয়েজে কোন নির্দিষ্ট Meaning বুঝায়। কী-ওয়ার্ড কে রিজার্ভ ওয়ার্ডও বলা হয়। কী-ওয়ার্ড গুলো ভ্যারিয়েবল অথবা ফাংশন এর নাম হিসেবে ব্যবহার করা যাবে না। কী-ওয়ার্ড যেমন: if, else ইত্যাদি। কী-ওয়ার্ড হচ্ছে এমন কিছু শব্দ যেগুলো আপনি আপনার ইচ্ছেমতো প্রোগ্রামিং এ ব্যবহার করতে পারবেন না। কারণ, প্রত্যেকটি প্রোগ্রামিং ল্যাংগুয়েজের নির্দিষ্ট কিছু কী-ওয়ার্ড আছে, যেগুলোকে ল্যাংগুয়েজে ডিফাইন করে দেয়া হয়েছে যে, ঐ কী-ওয়ার্ডসমূহের এর কাজ কি। প্রত্যেকটি কী-ওয়ার্ড এর নির্দিষ্ট কিছু কাজ আছে।

নিম্নে কতগুলো কী-ওয়ার্ড এর নাম দেওয়া হল

PHP Keywords :

abstract
and
array()
as
break
case
catch
class
clone
const
continue
declare
default
do
else
elseif
enddeclare
endfor
endforeach

endif
endswitch
endwhile
extends
final
for
foreach
function
global
goto
if
implements
interface
instanceof
namespace
new
or
private
protected
public
static
switch
throw
try
use
var
while
xor

c-27 পি.এইচ.পি কী-ওয়ার্ডস।

৫.১০ কনস্ট্যান্ট (Constant)

কনস্ট্যান্ট শব্দটির অর্থ হচ্ছে অপরিবর্তনীয়। যে কোন ধরনের প্রোগ্রামিং লজিক বুঝতে আপনি সংগ্রহ করতে পারেন বুকবিডি সিরিজের সি প্রোগ্রামিং বইটি। অনেক সময় প্রোগ্রাম চলাকালীন সময়ে অথবা পরে আমরা অনেক ভ্যারিয়েবল এর মান পরিবর্তন করি। আর যদি করতে না চাই তাহলে Constant কী-ওয়ার্ডটি ব্যবহার করা যেতে পারে।

Constant হচ্ছে আইডেন্টিফায়ার, যেটির Value প্রোগ্রাম চলাকালীন সময়ে পরিবর্তন করা যাবে না। ভ্যারিয়েবল এর মান যে কোন সময় পরিবর্তন করা যায়, কিন্তু constant এর মান একবারই assign করা যায় এবং পরবর্তীতে চাইলেও আর পরিবর্তন করা যায় না। যেমন Mathematics এ পাই এর মান সবসময় 3.14 এবং এর মান পরিবর্তন হয় না। সুতরাং, চাইলে এই মানটিকে constant এর সাথে তুলনা করা যায়। Constant যে কোন টাইপের value হতে পারে।

Constant তৈরি করতে হয় define() ফাংশন ব্যবহার করে। define() ফাংশন দুটি আরগুমেন্ট নেয়, প্রথমটি হচ্ছে constant এবং পরেরটি হচ্ছে value. Constant নাম case sensitive. Constant নামের সবগুলো লেটার আপার কেস হবে। নিম্নে Constant ডিক্লেয়ার করে দেখানো হলো

```
define ("MESSAGE", "welcomeBD");
```

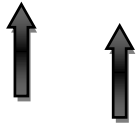


define ফাংশন। constant এ যে মানটি রাখতে চাই।

constant এর নাম, যে কোন একটি নাম হতে পারে তবে অবশ্যই Uppercase হতে হবে।

এখন যদি প্রশ্ন করা হয়, MESSAGE constant এর ভ্যালিউ/মান কি? তবে, উত্তর হবে welcome BD

```
define("VAL",20);
```



Constant Value of VAL
define("B",33);
echo B;

এখানে আউটপুটে আসবে 33, কারণ B এর মান হচ্ছে 33

নিচের প্রোগ্রামে MESSAGE এবং M নামে দুইটি Constant ভ্যারিয়েবল ডিক্লেয়ার করা হয়েছে এবং এদের Value হিসেবে Welcome to my World এবং 20 দেওয়া হয়েছে এবং এখানে যেহেতু echo MESSAGE এবং echo M লেখা হয়েছে, তাই আউটপুটে Welcome to my World এবং 20 (Welcome to my World20) আসবে।

```
<?php
class m {
    public function demo() {
        define("M", 20); // m is a variable the value of m cant chage latter
        echo M;
    }
}
$obj = new m;
$obj->demo();
?>
```

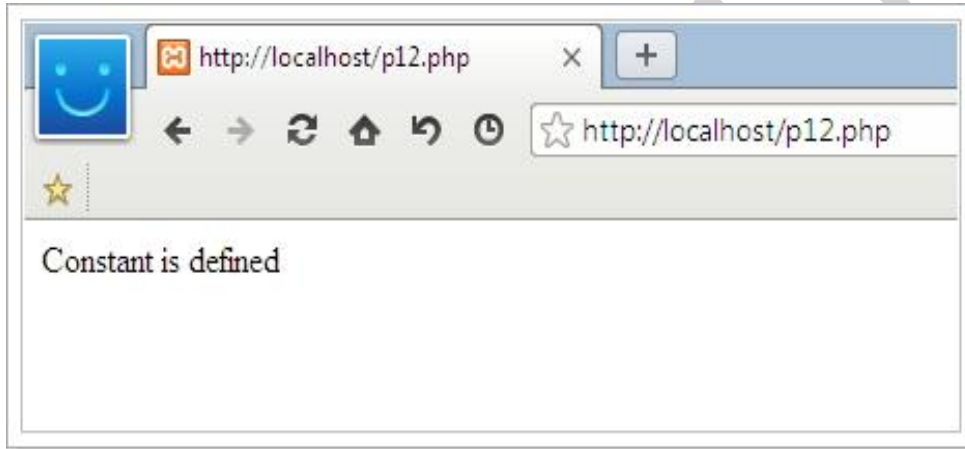
উপরের প্রোগ্রামটির Output: 20

নিচে MY_CONSTANT নামে একটি Constant ভ্যারিয়েবল ডিক্লেয়ার করে এর মান 36 দেওয়া হয়েছে। আবার এর সাথে if.....else conditional statement যুক্ত করা হয়েছে। নিচের প্রোগ্রামে constant ভ্যারিয়েবলে যদি MY_CONSTANT ডিফাইন করা হয়, তবে আউটপুট আসবে Constant is defined অন্যথায় আউটপুট আসবে Constant is not defined

অবজেক্ট অরিয়েটেড পিএইচপি

প্রোগ্রাম নং ২৭ (p27.php) Constant ভ্যারিয়েবলের মাধ্যমে Value অ্যাসাইন করে এদের আউটপুট দেখানো

```
<?php
define ("MY_CONSTANT", 36);
if (defined("MY_CONSTANT")) {
    echo "Constant is defined";
}
else {
    echo "Constant is not defined";
}
?>
```



চিত্রঃ ৫.১০ প্রোগ্রাম ২৭ এর আউটপুট

প্রোগ্রাম নং ২৮ (p28.php) Constant ভ্যারিয়েবলের সাথে if..else কন্ডিশনাল স্টেটম্যান্ট যুক্ত করে প্রোগ্রাম লিখা

```
<?php
// define your site name, since it does NOT change
// anywhere within your script.
define( 'SITE_NAME', 'Learning Journal' );
// define the current year, possibly to use in your copyright
// statement or for 'date' calculations
define( 'THIS_YEAR', date('Y') );
// even use existing constants to create other constants!
// e.g. adding the trademark symbol to your site name...
define( 'SITE_NAME_T', SITE_NAME.'&trade;' );
?>
```

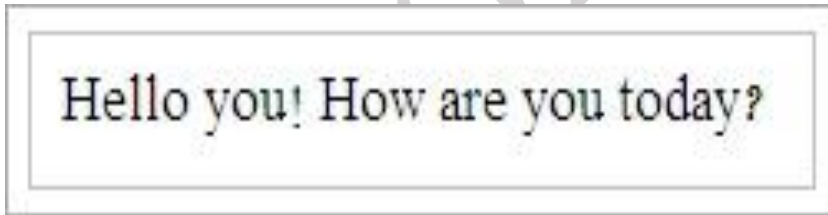
নিচের প্রোগ্রামে String, Integer, Boolean এবং Float টাইপের কনস্ট্যান্ট ভ্যারিয়েবল ডিক্লেয়ার করে দেখানো হয়েছে।

প্রোগ্রাম নং ২৯ (p29.php) Constant ভ্যারিয়েবল ডিক্লেয়ার করা

```
<?php
// String
define( 'AUTHOR', 'J de Silva' );
// Integer
define( 'COLUMNS', 3 );
// Boolean
define( 'SHOW_MENU', FALSE );
// Float
define( 'DISCOUNT_50', 0.5 );
?>
```

প্রোগ্রাম নং ৩০ (p30.php) বিভিন্ন ডেটা টাইপের কনস্ট্যান্ট ভ্যারিয়েবল ডিক্লেয়ার করা

```
<?php
//define a constant
define("GREETING","Hello you! How are you today?");
echo constant("GREETING");
?>
```



চিত্রঃ ৫.১০ প্রোগ্রাম ৩০ এর আউটপুট

নিচের প্রোগ্রামে MY_CONSTANT নামে কনস্ট্যান্ট ভ্যারিয়েবল ডিক্লেয়ার করা হয়েছে এবং এর Value হিসেবে The constant value দেওয়া হয়েছে।

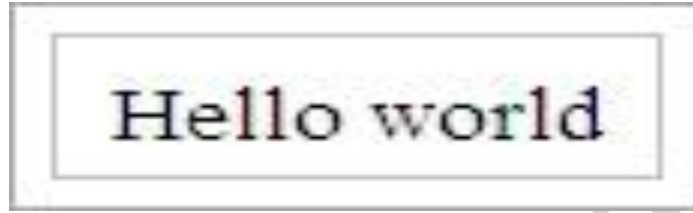
প্রোগ্রাম নং ৩১ (p31.php) কনস্ট্যান্ট ভ্যারিয়েবল ডিক্লেয়ার করে এর Value আউটপুটে দেখানো

```
<?php
define("MY_CONSTANT", "The constant value")
?>
```

নিচের প্রোগ্রামে কনস্ট্যান্ট ভ্যারিয়েবলের ভ্যালিউ Hello world দেওয়া হয়েছে এবং এর আউটপুট echo CONSTANT এর মাধ্যমে দেখানো হয়েছে। আবার echo Constant লেখায় এর আউটপুট আসবে ঠিকই, তবে উক্ত CONSTANT ভ্যারিয়েবলের মান না। কেননা, আগেই বলা হয়েছে, এটি কেস-সেনসিটিভ (UpperCase)। অর্থাৎ, এই প্রোগ্রামের আউটপুট আসবে Hello world.Constant

প্রোগ্রাম নং ৩২ (p32.php) কনস্ট্যান্ট ভ্যারিয়েবলের মাধ্যমে Value অ্যাসাইন করা

```
<?php
define("CONSTANT", "Hello world.");
echo CONSTANT; // outputs "Hello world."
echo Constant; //outputs "Constant" and issues a notice
?>
```



চিত্রঃ ৫.১০ প্রোগ্রাম ৩২ এর আউটপুট

৫.১১ কমেন্ট

কমেন্ট হচ্ছে একটি লাইন যেটি প্রোগ্রামের অংশ, কিন্তু প্রোগ্রাম execute করার সময় ঐ লাইনটিকে রিড করা হয় না। কমেন্ট এর মাধ্যমে সাধারণত কোড রিলেটেড কোন ইনফরমেশন দেওয়া থাকে, যেন অন্যরা কোড দেখে তা সহজে বুঝতে পারে।

সিংগেল লাইন কমেন্টের জন্য (//) ব্যবহার করা হয়। //echo "Beautiful" ← এটি হচ্ছে কমেন্ট লাইন। উক্ত প্রোগ্রামে আউটপুট হিসেবে Beautiful দেখাবে না, কারণ এটি কমেন্টের মধ্যে। আমরা চাইলে একের অধিক লাইনকে কমেন্টের মাধ্যমে execution বন্ধ রাখতে পারি।

একাধিক লাইনকে কমেন্ট এর আওতায় আনার জন্য শুরুতে front slash (/) এবং স্টার (*) সাইন ব্যবহার করতে হয়।

```
↓
/* echo "Dhaka";
$sp=6;
echo $sp;
*/ ← কমেন্টকে শেষ করতে হয় স্টার (*) এবং front slash (/) দিয়ে।
```

উপরের প্রোগ্রামে কমেন্টের ভিতরে যা রয়েছে, তা আউটপুটে কিছু আসবে না। অর্থাৎ এগুলো কোন কাজও করবে না।

নিচের প্রোগ্রামে echo "hello" এবং echo "there" লেখায় আউটপুট আসবে hello there। এখানে //this is a comment হচ্ছে একটি কমেন্ট, তাই এটি আউটপুটে আসবে না।

```
<?php
echo "hello";
//this is a comment
echo " there";
```

অবজেক্ট অরিয়েটেড পিএইচপি

?>

উপরের প্রোগ্রামটির : **Output:** hello there

প্রোগ্রাম নং ৩৩ (p33.php) কমেন্ট ব্যবহার করে প্রোগ্রাম লেখা

```
<?php
    echo "hello";
    /* Using this method
    you can create a larger block of text
    and it will all be commented out
    */
    echo "there";
?>
```

উপরের ৩৩ নং প্রোগ্রামটির : **Output:** hello there

নিচের প্রোগ্রামের কোন আউটপুট আসবেনা, কারণ এখানে //echo "hello"; লেখা হয়েছে এবং এটিকে লিখা হয়েছে কমেন্ট হিসেবে, আর যেহেতু কমেন্ট হিসেবে যা লিখা হয় তার আউটপুট ডিসপে হয় না, তাই এ প্রোগ্রামেরও কোন আউটপুট আসবে না।

প্রোগ্রাম নং ৩৪ (p34.php): কমেন্ট ব্যবহার করে প্রোগ্রাম লেখা

```
<?php
    //echo "hello";
?>
```

নিচের প্রোগ্রামে //echo "hello"; কমেন্ট হিসেবে ব্যবহার করায় এর কোন আউটপুট আসবেনা, কিন্তু যেহেতু আবার echo "there" লেখা হয়েছে এবং এটিকে কমেন্ট হিসেবে লিখা হয়নি, তাই এর আউটপুট আসবে there

প্রোগ্রাম নং ৩৫ (p35.php) শুধুমাত্র কমেন্ট ব্যবহার করা

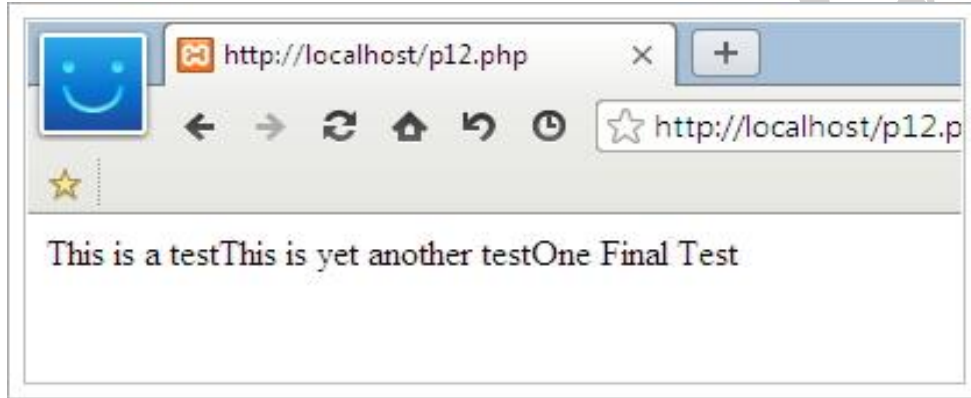
```
<?php
    //echo "hello";
    echo "there";
?>
```

উপরের ৩৫ নং প্রোগ্রামটির : **Output:** there

প্রোগ্রাম নং ৩৬ (p37.php) এই প্রোগ্রামের আউটপুট আসবে there

```
<?php
    echo 'This is a test'; // This is a one-line c++ style comment
```

```
/* This is a multi line comment
   yet another line of comment */
echo 'This is yet another test';
echo 'One Final Test'; # This is a one-line shell-style comment
?>
```



চিত্রঃ ৫.১১ প্রোগ্রাম ৩৬ এর আউটপুট

প্রোগ্রাম নং ৩৭ (p37.php)

```
<?php
/* echo 'This is a test'; /* This comment will cause a problem */
*/
?>
```

প্রোগ্রাম নং ৩৮ (p38.php)

অন্য আরেকটি অপশন ব্যবহার করেও আমরা কमेंট করতে পারি। হেশ (#) সাইন ব্যবহার করেও কमेंট করা যায়।

```
#echo "Girl";
```

সুতরাং, এই প্রোগ্রামের আউটপুটে Girl আসবেনা।

নিচের প্রোগ্রামে # ব্যবহার করে কमेंট লেখা হয়েছে।

```
<?php
echo "hello";
#this is a comment
echo " there";
?>
```

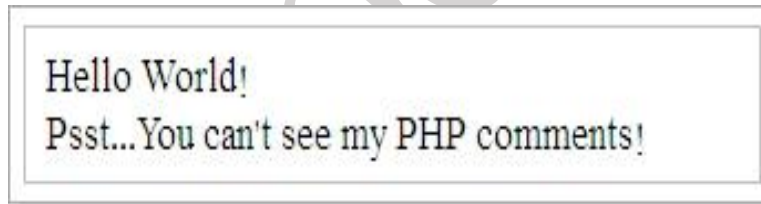



চিত্রঃ ৫.১১ প্রোগ্রাম ৩৮ এর আউটপুট

নিচের প্রোগ্রামে #echo "I don't do anything either" কে কমেন্ট হিসেবে ব্যবহার করা হয়েছে, তাই এখানে echo লিখা সত্ত্বেও এর কোন আউটপুট আসবে না। এই প্রোগ্রামের আউটপুট আসবে Hello World! Psst...You can't see my PHP comments!

প্রোগ্রাম নং ৩৯ (p39.php) # ব্যবহার করে কমেন্ট লেখা

```
<?php
    echo "Hello World!"; // This will print out Hello World!
    echo "<br />Psst...You can't see my PHP comments!"; // echo "nothing";
    // echo "My name is Humperdinkle!";
    # echo "I don't do anything either";
?>
```



চিত্রঃ ৫.১১ প্রোগ্রাম ৩৯ এর আউটপুট

৫.১২ সংক্ষিপ্ত অধ্যায়

> প্রোগ্রামিং এ প্রোগ্রাম লিখা হয় ভ্যালুকে (Data) প্রসেস করার জন্য। প্রোগ্রামিং এ বিভিন্ন ধরনের ডেটা থাকে যেমন ইনটিজার (৫,১১,১০৯ ইত্যাদি) ক্যারেক্টার ("a", "c", "z", "p" ইত্যাদি) স্ট্রিং ("book", "bookbd", "Dhaka", "green" ইত্যাদি) ভ্যালুকে আমরা ভ্যারিয়েবল এ রাখি। ভ্যারিয়েবল যেমন a, dk, cz ইত্যাদি। dk=2 তাহলে বলতে পারি dk ভ্যারিয়েবল এর মান হচ্ছে ২।

```
> class bd
{
```

```
    public $d= "p";
    public Function dom() ← নিয়ম ফাংশন এর নামের পর দু'টো প্রথম বন্ধনী দিতে হয়
    { ← ফাংশন এর ব্রাকেট শুরু
```

```
$name= "php";  
echo $name;  
$price = 200;  
echo $price;  
} ← ফাংশন এর ব্রাকেট শেষ  
}
```

> উপরের ক্লাসটির অবজেক্ট দিয়ে dom ফাংশনটিকে কল করলে আউটপুট প্রদর্শিত হবে php এবং ২০০।

> পিএইচপিতে এমন নির্দিষ্ট শব্দ (word) আছে যে গুলো প্রোগ্রামার (ব্যবহারকারী) চাইলে তার ইচ্ছে মত যে কোন জায়গায় ব্যবহার করতে পারবেন না। আর এটিই হচ্ছে কী-ওয়ার্ড। যেমন আমি একটি ফাংশন এর নাম পাবলিক (public) লিখতে পারবো না কারণ এটি হচ্ছে একটি কী-ওয়ার্ড যেমন: use, while, class, and, exit ইত্যাদি।

৫.১৩ প্রশ্ন

১. প্রোগ্রামিং এ ভ্যালু বলতে কি বুঝ?
২. ভ্যারিয়েবল কি?
৩. ভ্যালু এবং ভ্যারিয়েবল এর মধ্য পার্থক্য লিখ।
৪. পিএইচপিতে ভ্যারিয়েবল ডিক্লেয়ার করার নিয়ম গুলো লিখ।
৫. বিভিন্ন ধরনের ডেটা টাইপ গুলো উদাহরণ সহ বর্ণনা কর।
৬. কনস্ট্যান্ট কি এবং কিভাবে ডিক্লেয়ার করতে হয়।
৭. কমেণ্ট কেন ব্যবহার করা হয়।
৮. একটি ক্লাস লিখ এবং যেটি প্রিন্ট (Display) করে Green Dhaka। তারপর ঐ ক্লাসের ফাংশনটিকে কল কর। (এখানে মনে রাখতে হবে কোন ফাংশনের ভিতর আউটপুটের জন্য কোডটি লিখতে হবে।
৯. একটি ক্লাস Mango ঐ ক্লাসের ভিতর একটি ভ্যারিয়েবল p তে ২৩ অ্যাসাইন করা হল। তার Mango ক্লাসের অবজেক্ট তৈরী করে p এর মানকে আউটপুট প্রদর্শন করতে হবে।

QUICK & EASY WAY

Your Professional Trainer...

নিজে নিজে এসইও শিখুন

অ্যাডভান্স

সার্চ ইঞ্জিন অপটিমাইজেশন

From Advance to Professional

রচনা ও সম্পাদনা

বুকবর্ডি মিরিজ...

কী-ওয়ার্ড অ্যানালাইসিস * সার্চ ইঞ্জিন সিক্রেট * এসইও টুলস * মেইল চিফ্প
* গুগলের নতুন এসইও স্ট্র্যাটেজি * লিংক তৈরি * এসইও ট্রাবল শুটিং



কিন্ডারে সবচেয়ে বড় সার্চ ইঞ্জিনকে কন্ট্রোল করে বিশ্বের মাঝে আপনার ওয়েব
মাইটকে বেশি পরিচিতি করা যায় এ পুরো বিস্মাটিক বইটি পড়ে খুব সহজে বুঝবেন।
অল্প শীক্ষিত এবং টেকনিক্যাল পার্টক খুব সহজে কন্ট্রোল শিখতে পারবেন।



All Updated Tools & Techniques of SEO

ডাউনলোড

bookbd.info

অধ্যায়-৬ অপারেটরস এ্যান্ড এক্সপ্রেশন

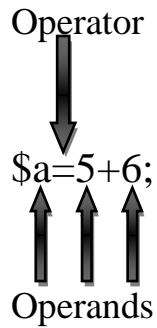
৬. অপারেটর, এক্সপ্রেশন, অপারেটর টাইপস সম্পর্কে আলোচনা -----
- ৬.১ ভূমিকা : -----
- ৬.২ অপারেটর (Operator) -----
- ৬.৩ এক্সপ্রেশন (Expression) -----
- ৬.৪ অপারেটর টাইপস (Operator Types) -----
- ৬.৫ অ্যারিথমেটিক অপারেটর (Arithmetic Operators) -----
- ৬.৬ অ্যাসাইনমেন্ট অপারেটরস (Assignment Operators) -----
- ৬.৭ কম্প্যারিজন অপারেটরস (Comparision Operators) -----
- ৬.৮ লজিক্যাল অপারেটর (Logical Operators) -----
- ৬.৯ অধ্যায় সারসংক্ষেপঃ -----
- ৬.১০ প্রশ্নঃ -----

৬.১ ভূমিকা :

কতগুলো সিম্বলকে একত্রে আমরা প্রোগ্রামিং এ এক্সপ্রেশন বলে থাকি যেমন $a=2$ এখানে এটি হচ্ছে একটি এক্সপ্রেশন। Expression কে ভ্যালু, ভ্যারিয়েবল এবং অপারেটর এর একটি কম্বিনেশন (Combination) ও বলা হয়ে থাকে। অপারেটর হচ্ছে এমন একটি জিনিস যেটি ভ্যালুকে পরিবর্তন করতে পারে। যেমন $a=3$ এখানে a এর মান 3 কারণ ইকুয়াল (=) অপারেটর দ্বারা এর মান পরিবর্তন করা হল। তাহলে বলা যায় অপারেটর হচ্ছে এক ধরনের সিম্বল (Symbol) যার একটি নির্দিষ্ট কাজ আছে। অপারেটর যেমন প্লাস (+), মাল্টিপিকেশন (*), ডিভিশন (/) ইত্যাদি।

৬.২ অপারেটর (Operator) :

অপারেটর হচ্ছে এক ধরনের সিম্বল (symbol) যেটি কম্পিউটারকে নির্দেশ করে কোন ধরনের Mathematical অথবা Logical ম্যানিপুলেশন (অপারেশন) করতে হবে। প্রোগ্রামে অপারেটর ব্যবহার করা হয় ডেটা এবং ভ্যারিয়েবলকে ম্যানিপুলেট (বিভিন্ন কাজ পরিচালনা) করার জন্য। অপারেটর এর সাহায্যে আমরা এক বা একাধিক মান থেকে অন্য নতুন কোন ভ্যালিউ তৈরি করতে পারি।

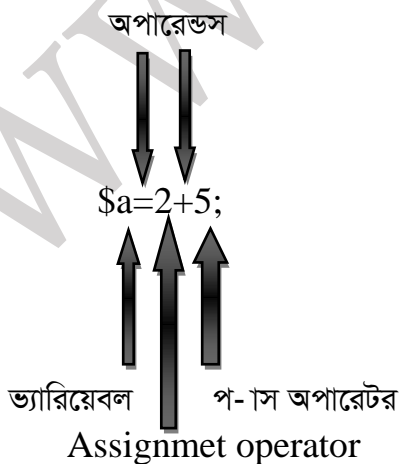


ভ্যারিয়েবল a তে 5 এবং 6 এর যোগফল 11

অপারেটর হচ্ছে একটি symbol যেটি এক বা একাধিক অপারেন্ড এর উপর কোন ধরনের অপারেশন চালায়।

৬.৩ এক্সপ্রেশন (Expression) :

এক্সপ্রেশন হচ্ছে কিছু code (piece of code) যেটি প্রডিউস করে কোন রেজাল্ট অথবা মান। প্রোগ্রামিং ল্যাংগুয়েজে এক্সপ্রেশন-ভ্যালিউ, ভ্যারিয়েবল, কনস্ট্যান্ট, অপারেটর ইত্যাদিকে একত্র করে।



অবজেক্ট অরিয়েটেড পিএইচপি

পুরো লাইন কোডটি হচ্ছে এক্সপ্রেশন।

অর্থাৎ প্রোগ্রামিং এর কোন একটি statement কে আমরা এক্সপ্রেশন বলতে পারি। যেমন :

`a=b+3;`

`z++;`

`300>(8*k);`

`a*b+c;`

`(a*b)+c;`

`(a*b)/c;`

৬.৪ অপারেটর টাইপস (Operator Types) :

প্রোগ্রামিং এ অপারেটর গুলোকে সাধারণত কতগুলো ভাগে ভাগ করা হয়েছে।

1. Arithmetic operators (অ্যারিথমেটিক অপারেটরস)
1. Relational operators (রিলেশনাল অপারেটরস)
2. Logical operators (লজিক্যাল অপারেটরস)
3. Assignment operators (অ্যাসাইনমেন্ট অপারেটরস)
8. Increment and Decrement operators (ইনক্রিমেন্ট এবং ডিক্রিমেন্ট অপারেটরস)
৫. Conditional operators (কন্ডিশনাল অপারেটরস)
6. Bitwise operators (বিটওয়াইজ অপারেটরস)
9. Special operators (স্পেশাল অপারেটরস)

৬.৫ অ্যারিথমেটিক অপারেটর (Arithmetic Operators) :

নিম্নে Arithmetic অপারেটর গুলো আলোচনা করা হল :

+ অ্যাডিশন (Addition)

`x=2+5;`

Addition

নিচের প্রোগ্রামে `$a=5+6;` এর মাধ্যমে 5 এবং 6 কে যোগ করা হয়েছে। এখানে + Addition অপারেটর ব্যবহার করা হয়েছে। এর আউটপুট আসবে 11

```
<?php
Class op{
    $a=5+6;
}
$O=new op;
echo $O->a;
?>
```

অবজেক্ট অরিয়েটেড পিএইচপি

প্রোগ্রাম নং ৪০ (p40.php) Addition অপারেটরের মাধ্যমে দুটি সংখ্যা যোগ করা

উপরের প্রোগ্রামটি রান করলে আমরা আউটপুটে 11 পাবো।

```
<?php
    $a=6;
    echo $a+5;
?>
```



চিত্রঃ ৬.৫ প্রোগ্রাম p40 এর আউটপুট

- সাবট্রাকশন (Subtraction)

```
$x=6-2;
```

Subtraction



এখানে x এর মান হবে 4

```
$a=8;
```

```
$b=$a-3;
```

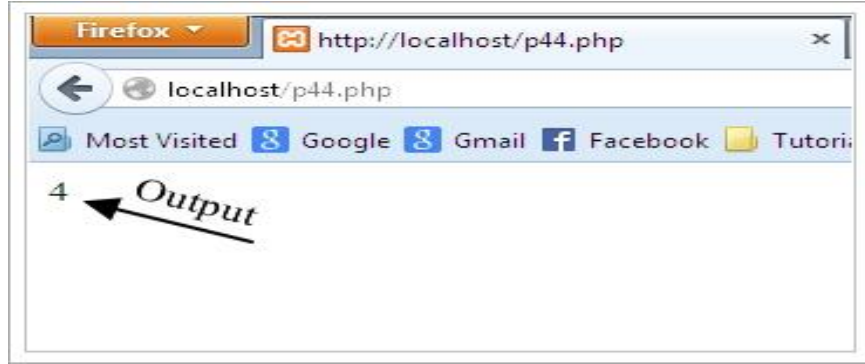
b এর মান হবে 5

```
$c=$a-$b;
```

তাহলে c এর মান হবে 3 কারণ, a এর মান 8 আর তা থেকে b এর মান 5 বাদ গেলে 3 থাকে। নিচের প্রোগ্রামে ভ্যারিয়েবল \$a=12-8 লেখা হয়েছে অর্থাৎ এখানে Subtraction (-) অপারেটরের মাধ্যমে 12 থেকে 8 বিয়োগ করে, বিয়োগফল echo \$a এর মাধ্যমে দেখানো হচ্ছে।

প্রোগ্রাম নং ৪১ (p41.php) Addition অপারেটরের মাধ্যমে যোগ করা

```
<?php
    $a=12-8;
    echo $a;
?>
```

চিত্রঃ ৬.৫ প্রোগ্রাম p41 এর আউটপুট

প্রোগ্রাম নং ৪২ (42.php) Subtraction অপারেটর এর ব্যবহার

```
<?php
Class cal
{
    Public function cum()
    {
        $a=12-8;
        $c=$a-2;
        echo $c;
        $d= $a - $c;
        echo $d;
    }
}
P=new cal;
p->cum();
?>
```

* মাল্টিপ্লিকেশন (Multiplication)

```
$p=3*2;
$p=6;
$m=4;
$x=$p*$m;
```

Multiplication



এখানে, x এর মান হবে 24 কারণ, p এর মান 6 এবং m এর মান 4 সুতরাং 4 এবং 6 এর গুনফল 24. নিচের প্রোগ্রামে ভ্যারিয়েবল \$a=8*2; লিখা হয়েছে অর্থাৎ এখানে Multiplication (*) অপারেটরের মাধ্যমে 8 এর সাথে 2 গুন করে, গুনফল echo \$a এর মাধ্যমে দেখানো হচ্ছে।

প্রোগ্রাম নং ৪৩ (p43.php) Subtraction অপারেটর এর ব্যবহার

```
<?php
    $a=8*2;
    echo $a;
?>
```



চিত্রঃ ৬.৫ প্রোগ্রাম p43 এর আউটপুট

প্রোগ্রাম নং ৪৪ (p44.php) Multiplication অপারেটর এর ব্যবহার

```
<?php
Class Mul {
    public function multification() {
        $a=3*2;
        $m=5;
        $p=$a * $m;
        echo $p;
    }
}
P=new Mul;
p-> multification();
?>
```

/ ডিভিশন (Division)

```
$x=15/5;
x এর মান হবে 3
$a=12;
$b=3;
$p=$a/$b
p এর মান হবে 4
```

নিচের প্রোগ্রামে ভ্যারিয়েবল \$a=18/3 লিখা হয়েছে, অর্থাৎ এখানে Division (/) অপারেটরের মাধ্যমে 18 কে 3 দ্বারা ভাগ করে, ভাগফল echo \$a এর মাধ্যমে দেখানো হচ্ছে।

প্রোগ্রাম নং ৪৫ (p45.php) Multiplication অপারেটর এর ব্যবহার

```
<?php
Class Div {
    public function division() {
        $a=18/3;
        echo $a;
    }
}
P=new Div;
$P-> division();
?>
```

প্রোগ্রাম নং ৪৬ (p46.php) Division অপারেটর এর ব্যবহার

```
<?php
$a=25;
$b=5;
$x=$a/$b;
echo $x;
?>
```

উপরের ৪৬ নং প্রোগ্রামের Output: 5

% মডিউলাস (Modulus)

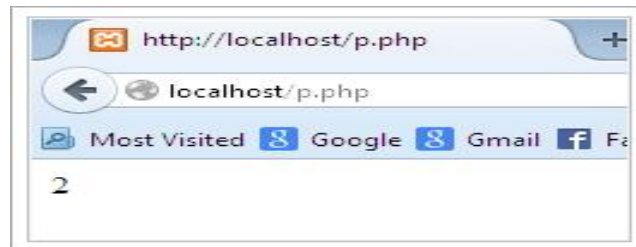
```
$m=11%2;
```

এখানে, m এর মান হবে 1. কারণ মডিউলাস মানে হচ্ছে ভাগশেষ। সুতরাং, 11 কে 2 দিয়ে ভাগ করলে, ভাগশেষ থাকে 1.

নিচের প্রোগ্রামে ভ্যারিয়েবল \$a=20%3; লিখা হয়েছে, অর্থাৎ এখানে Modulus (%) অপারেটরের মাধ্যমে 20 কে 3 দ্বারা মডিউলাস করে, ভাগশেষ echo \$a এর মাধ্যমে দেখানো হচ্ছে।

প্রোগ্রাম নং ৪৭ (p47.php): Division অপারেটর এর ব্যবহার

```
<?php
$a=20%3;
echo $a;
?>
```



চিত্রঃ ৬.৫ প্রোগ্রাম p47 এর আউটপুট

অবজেক্ট অরিয়েটেড পিএইচপি

প্রোগ্রাম নং ৪৮ (p48.php) Modulus অপারেটর এর ব্যবহার

```
<?php
    $a=29;  $b=5;
    $x=$a%$b;
    echo $x;
?>
```

চিত্র উপরের ৪৮ নং প্রোগ্রামের : Output: 4

```
<?php
    Class Mod {
        public function division(){
            $a=18/3;
            echo $a;
        }
    }
    $p=new Mod;
    $p->division();
?>
```

উপরের প্রোগ্রামের : Output: 6

++ ইনক্রিমেন্ট (Increment)

```
$x=6;
$x++;
এখানে x এর মান হবে 7 কারণ increment 1 করে বাড়ে।
$p=2;
$m=$p++;
```

m এর মান হবে 3 কারণ p এর মান ছিল 2 তার সাথে এক যোগ হয়ে এখন 3 হয়েছে।

নিচের প্রোগ্রামে \$a নামে একটি ভ্যারিয়েবল ডিক্লেয়ার করে এর ভ্যালিউ 2 দেওয়া হয়েছে, তারপর \$a++ এর মাধ্যমে এর ভ্যালিউকে 1 বাড়িয়ে দেওয়া হল। প্রথমে \$a এর ভ্যালিউ ছিল 2, কিন্তু \$a++ ব্যবহার করায় এর ভ্যালিউ বেড়ে হল 3. সুতরাং, এখন এর আউটপুট আসবে 3.

প্রোগ্রাম নং ৪৯ (49.php) Modulus অপারেটর এর ব্যবহার

```
<?php
    $a=2;
    $a++;
    echo $a;
```

অবজেক্ট অরিয়েটেড পিএইচপি

?>

উপরের 89 নং প্রোগ্রামের : Output: 3

প্রোগ্রাম নং ৫০ (p50.php) Increment অপারেটর এর ব্যবহার

```
<?php
    $a=8;
    $p=$a++;
    echo $p;
?>
```

উপরের ৫০ নং প্রোগ্রামের : Output: 8

```
<?php
Class Inc {
    public function increment(){
        $a=8;
        $d=$a++;
        echo $d;
    }
}
$p=new Inc;
$p-> increment();
?>
```

উপরের প্রোগ্রামের : Output: 8

-- ডিক্রিমেন্ট (Decrement)

(decrement) ডিক্রিমেন্ট অপারেটর কোন Value এর মান এক এক করে কমায় (কমিয়ে দেয়)।

```
$c=5; $c--;
```

এখানে, c এর মান হবে 4।

```
$z=9; $p=$z--;
```

p এর মান হবে 8 কারণ 9 থেকে 1 কমেছে অর্থাৎ subtract হয়েছে।

প্রোগ্রাম নং ৫১ (p51.php) Increment অপারেটর এর ব্যবহার

```
<?php
Class Dec {
    public function decrement(){
        $c=5;
```

```

        $c--;
    }
}
$p=new Dec;
$p-> decrement();
?>

```

অ্যারিথমেটিক অপারেটরস (Arithmetic Operators)

Operator	Description	Example	Result
+	Addition	x=2 x+2	4
-	Subtraction	x=2 5-x	3
*	Multiplication	x=4 x*5	20
/	Division	15/5 5/2	3 2.5
%	Modulus (division remainder)	5%2 10%8 10%2	1 2 0
++	Increment	x=5 x++	x=6
--	Decrement	x=5 x--	x=4

নিচের প্রোগ্রামে বিভিন্ন Arithmetic অপারেটর অর্থাৎ Addition, Subtracction, Multiplication, Division, Modulus ব্যবহার করে প্রোগ্রাম লিখা হল।

```

<?php
$addition = 2 + 4;
$subtraction = 6 - 2;
$multiplication = 5*3;
$division = 15/3;
$modulus = 5%2;
echo "Perform addition: 2 + 4 = ".$addition."<br />";
echo "Perform subtraction: 6 - 2 = ".$subtraction."<br />";
echo "Perform multiplication: 5 * 3 = ".$multiplication."<br />";

```

```
echo "Perform division: 15 / 3 = ".$division."<br />";  
echo "Perform modulus: 5 % 2 = ".$modulus . " ". Modulus is the  
remainder after the division operation has been performed. In this case it was 5 / 2,  
which has a remainder of 1.";
```

?>

নিচে Post Increment এবং Pre Increment ব্যবহার করে একটি প্রোগ্রাম লিখা হল :

প্রোগ্রাম নং ৫২ (p52.php) বিভিন্ন Arithmetic অপারেটর এর ব্যবহার

```
<?php
```

```
$x = 4;
```

```
echo "The value of x with post-plusplus = " . $x++;
```

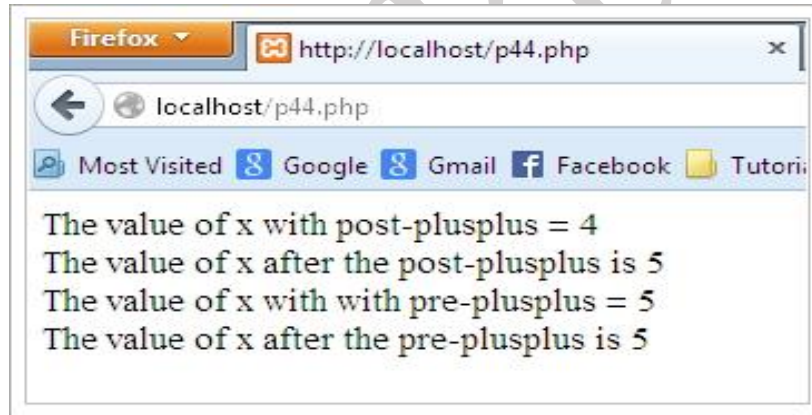
```
echo "<br /> The value of x after the post-plusplus is " . $x;
```

```
$x = 4;
```

```
echo "<br />The value of x with with pre-plusplus = " . ++$x;
```

```
echo "<br /> The value of x after the pre-plusplus is " . $x;
```

?>



চিত্রঃ ৬.৫ প্রোগ্রাম p52 এর আউটপুট

প্রোগ্রাম নং ৫৩ (p53.php) Post Increment এবং Pre Increment এর ব্যবহার

```
<?php
```

```
$add = 6 + 4;
```

```
$sub = 6 - 4;
```

```
$multi = 6 * 4;
```

```
$div = 6 / 3;
```

```
$mod = 10 % 2;
```

```
echo " 6 + 4 = ".$add."<br />";
```

```
echo " 6 - 4 = ".$sub."<br />";
```

```
echo " 6 * 4 = ".$multi."<br />";
```

```
echo " 6/3 = ".$div."<br />";
```

```
echo " 10%2 = ".$mod."<br />";  
?>
```



চিত্রঃ ৬.৫ শ্রেণী p53 এর আউটপুট

- $\$a += \b ; is the same as $\$a = \$a + \$b$;
- $\$a -= \b ; is the same as $\$a = \$a - \$b$;
- $\$a *= \b ; is the same as $\$a = \$a * \$b$;
- $\$a /= \b ; is the same as $\$a = \$a / \$b$;
- $\$a \% = \b ; is the same as $\$a = \$a \% \$b$;

c-59

শ্রেণী নং ৫৪ (p54.php) বিভিন্ন Arithmetic অপারেটর এর ব্যবহার

```
<html>  
<head><title>Arithmetical Operators</title></head>  
<body>  
<?php  
  $a = 42;  
  $b = 20;  
  $c = $a + $b;  
  echo "Addition Operation Result: $c <br/>";  
  $c = $a - $b;  
  echo "Substraction Operation Result: $c <br/>";  
  $c = $a * $b;  
  echo "Multiplication Operation Result: $c <br/>";  
  $c = $a / $b;  
  echo "Division Operation Result: $c <br/>";  
  $c = $a % $b;  
  echo "Modulus Operation Result: $c <br/>";  
  $c = $a++;
```



```
echo "Increment Operation Result: $c <br/>";  
$c = $a--;  
echo "Decrement Operation Result: $c <br/>";  
?>  
</body></html>
```



চিত্রঃ ৬.৫ প্রোগ্রাম p54এর আউটপুট

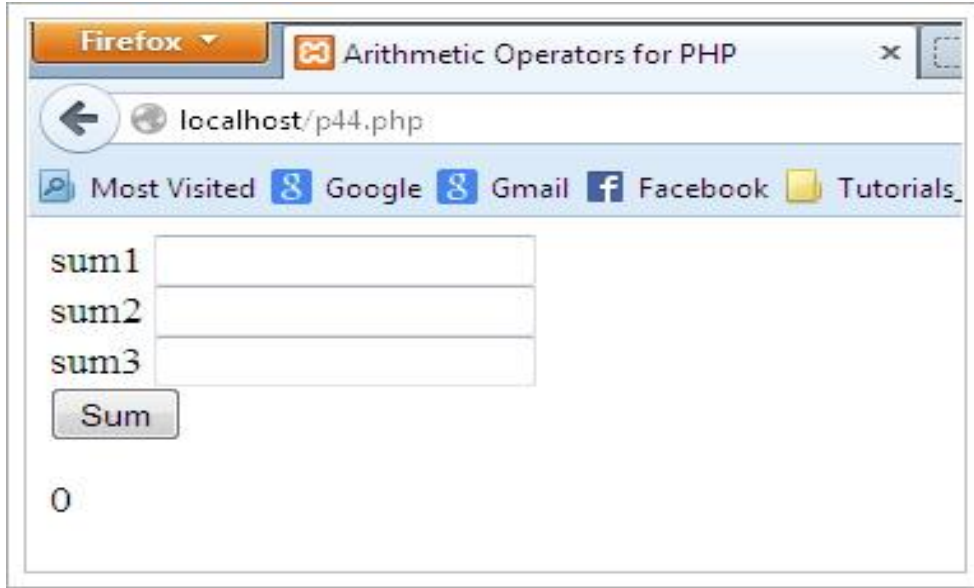
নিচের প্রোগ্রামে ইউজার পি.এইচ.পি ফর্ম (ফর্ম ইউজার ইনপুট অর্থাৎ কী-বোর্ড থেকে ইনপুট নিতে ব্যবহৃত হয়) এর মাধ্যমে তিনটি সংখ্যা কী-বোর্ড থেকে ইনপুট দিবে এবং ইউজারের দেওয়া তিনটি সংখ্যার যোগফল আউটপুটে ডিসপে- করবে।

প্রোগ্রাম নং ৫৫ (p55.php) বিভিন্ন Arithmetic অপারেটর এর ব্যবহার

```
<html>  
<head><title>Arithmetic Operators for PHP</title> </head>  
<Body>  
<form name="form1" method="post" action="example.php">  
sum1 <input name="sum1" type="text"><br>  
sum2 <input name="sum2" type="text"><br>  
sum3 <input name="sum3" type="text"><br>  
<input type="submit" name="submit" value="Sum"><br>  
</form>  
<?php  
$sum1=$_POST['sum1'];  
$sum2=$_POST['sum2'];  
$sum3=$_POST['sum3'];  
$total_sum=$sum1+$sum2+$sum3;  
echo $total_sum;
```

অবজেক্ট অরিয়েটেড পিএইচপি

```
?>  
</body> </html>
```

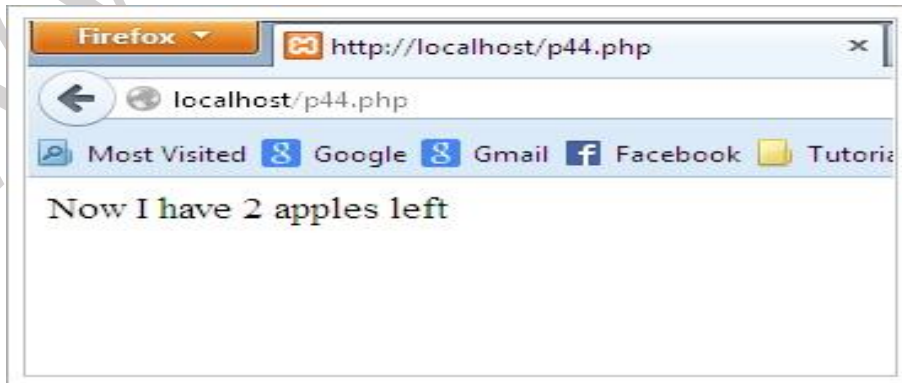


চিত্রঃ ৬.৫ প্রোগ্রাম p55 এর আউটপুট

(উপরে "example.php" এর জায়গায় আপনি আপনার ফাইলের নাম দিবেন। অর্থাৎ, যে নামে আপনি উক্ত ফাইলটি সেভ করেছেন)।

প্রোগ্রাম নং ৫৬ (p56.php) Arithmetic অপারেটর এর ব্যবহার

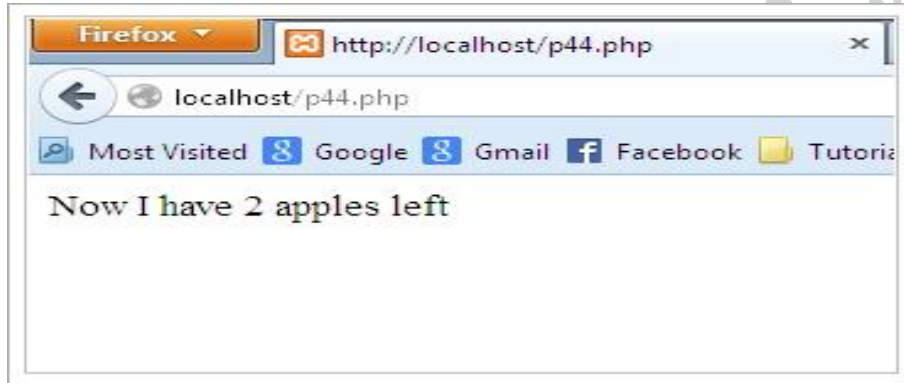
```
<?php  
$applesBought = 5;  
$applesSold = 3;  
echo "Now I have " . $applesBought - $applesSold . " apples left";  
?>
```



চিত্রঃ ৬.৫ প্রোগ্রাম p56 এর আউটপুট

প্রোগ্রাম নং ৫৭ (p57.php) Arithmetic অপারেটর এর ব্যবহার

```
<?php
    $applesBought = 5;
    $applesSold = 3;
    echo "Now I have " . ($applesBought - $applesSold) . " apples left";
    // Outputs:
    // Now I have 2 apples left
?>
```



চিত্রঃ ৬.৫ প্রোগ্রাম p57 এর আউটপুট

প্রোগ্রাম নং ৫৮ (p58.php) Arithmetic অপারেটর এর ব্যবহার।

```
<?php
// PHP has 5 primary math operators: addition, subtraction, multiplication,
division, and modulus
// These should be self explanatory with the exception of modulus. Modulus gives
you the remainder of
// a division. For example, the modulus of 5 and 2 is 1, since 1 is the remainder.
Here are examples:
echo 3 + 1; // Addition, output would be 4
echo 3 - 1; // Subtraction, output would be 2
echo 3 * 2; // Multiplication, output would be 6
echo 6 / 2; // Division, output would be 3
echo 7 % 4; // Modulus, output would be 3
// When dividing numbers, PHP will automatically give you a decimal number if
there is a remainder.
// Notice above that I did not surround the numbers with quotes. Strictly speaking,
a number in a programming
// language that is surrounded by quotes is no longer a number, but a string. PHP
will, however, still perform
```

// mathematical operations on numbers inside of quotes, but it is recommended to avoid doing this.

// Here is an example of this:

```
echo '7' * '3';
```

// Will have the same output as this:

```
echo 7 * 3;
```

```
?>
```



চিত্রঃ ৬.৫ শ্রেণী p58 এর আউটপুট

শ্রেণী নং ৫৯ (p59.php) বিভিন্ন Arithmetic অপারেটর এর ব্যবহার

```
<?php
```

```
$x=100;
```

```
$y=60;
```

```
echo "The sum of x and y is : ". ($x+$y) . "<br />";
```

```
echo "The difference between x and y is : ". ($x-$y) . "<br />";
```

```
echo "Multiplication of x and y : ". ($x*$y) . "<br />";
```

```
echo "Division of x and y : ". ($x/$y) . "<br />";
```

```
echo "Modulus of x and y : ". ($x%$y) . "<br />";
```

```
?>
```



চিত্রঃ ৬.৫ শ্রেণী p59 এর আউটপুট

প্রোগ্রাম নং ৬০ (p60.php) বিভিন্ন Arithmetic অপারেটর এর ব্যবহার

```
<?php
$num1 = 50;
$num2 = 30;
//Addition
$sum = $num1 + $num2;
echo 'Addition = '. $sum. "<br>";
//Subtraction
$difference = $num1 - $num2;
echo 'Subtraction = '. $difference. "<br>";
//Multiplication
$product = $num1*$num2;
echo 'Multiplication = '. $product ."<br>";
//Division
$quotient = $num1 / $num2;
echo 'Division = '. $quotient. "<br>";
//Modulus
$remainder = $num1 % $num2;
echo 'Modulus = '. $remainder ."<br>";
?>
```



চিত্রঃ ৬.৫ প্রোগ্রাম p60 এর আউটপুট

প্রোগ্রাম নং ৬১ (p61.php) বিভিন্ন Arithmetic অপারেটর এর ব্যবহার

```
<?php
// addition
$var1 = 4;
$var2 = 5;
```

অবজেক্ট অরিয়েটেড পিএইচপি

```
$total = $var1 + var2;  
echo "Total = " . $total;  
// Total = 9  
?>
```



চিত্রঃ ৬.৫ প্রোগ্রাম p61 এর আউটপুট

প্রোগ্রাম নং ৬২ (p62.php) **Arithmetic** অপারেটর এর ব্যবহার

```
<?php  
// subtraction  
$var1 = 8;  
$var2 = 5;  
$total = $var1 - var2;  
echo "Total = " . $total;  
// Total = 3  
?>
```



চিত্রঃ ৬.৫ প্রোগ্রাম p62 এর আউটপুট

প্রোগ্রাম নং ৬৩ (p63.php) **Arithmetic** অপারেটর এর ব্যবহার

```
<?php  
// Division  
$var1 = 15; $var2 = 5;
```

অবজেক্ট অরিয়েটেড পিএইচপি

```
$total = $var1 / var2;  
echo "Total = " . $total;  
// Total = 3  
?>
```



চিত্রঃ ৬.৫ প্রোগ্রাম p63 এর আউটপুট

প্রোগ্রাম নং ৬৪ (p64.php) Arithmetic অপারেটর এর ব্যবহার

```
<?php  
// Modulus  
$var1 = 15;  
$var2 = 5;  
$total = $var1 % var2;  
echo "Total = ".$total;  
// Total = 0  
?>
```



চিত্রঃ ৬.৫ প্রোগ্রাম p64 এর আউটপুট

প্রোগ্রাম নং ৬৫ (p65.php) Arithmetic অপারেটর এর ব্যবহার

```
< ?php  
$num1 = 50;  
$num2 = 30;
```

```
//Addition
$num1 = $num1 + $num2;
//Shorthand for this is..
$num1 += $num2;
//Subtraction
$num1 = $num1 - $num2;
//Shorthand for this is..
$num1 -= $num2;
//Multiplication
$num1 = $num1 * $num2;
//Shorthand for this is..
$num1 *= $num2;
//Division
$num1 = $num1 / $num2;
//Shorthand for this is..
$num1 /= $num2;
//Modulus
$num1 = $num1 % $num2;
//Shorthand for this is..
$num1 %= $num2;
?>
```



চিত্রঃ ৬.৫ প্রোগ্রাম p65 এর আউটপুট

প্রোগ্রাম নং ৬৬ (p66.php) বিভিন্ন Arithmetic অপারেটর এর ব্যবহার

```
<!DOCTYPE html PUBLIC "-//W3C//DTD/XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml11-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head><title>A Web Page</title></head>
<body>
<p>// The addition operator
```



```
<?php
$sum = 5 + 2;
$sum = 7;
$newsum = $sum + 4;
echo "The sum is " . $newsum
//The subtraction operator
$difference = $newsum - 2;
echo "The difference is " . $difference;
//The multiplication operator
$product = $difference * 3;
echo "The product is " . $product;
//The division operator
$quotient = $product / $difference;
echo "The quotient is " . $quotient;

//The Increment operator
$quotient++;
echo $quotient;
//The decrement operator
$quotient--;
echo $quotient;
?>
</p>
</body></html>
```



চিত্রঃ ৬.৫ প্রোগ্রাম p66 এর আউটপুট

প্রোগ্রাম নং ৬৭ (p67.php) বিভিন্ন Arithmetic অপারেটর এর ব্যবহার

```
<?php
$num1 = 4;
$num2 = 5;
```

অবজেক্ট অরিয়েটেড পিএইচপি

```
$num3 = 2;  
$answer = $num1 * $num2 - $num3;  
echo $answer;  
?>
```

উপরের ৬৭ নং প্রোগ্রামের : **Output: 18**

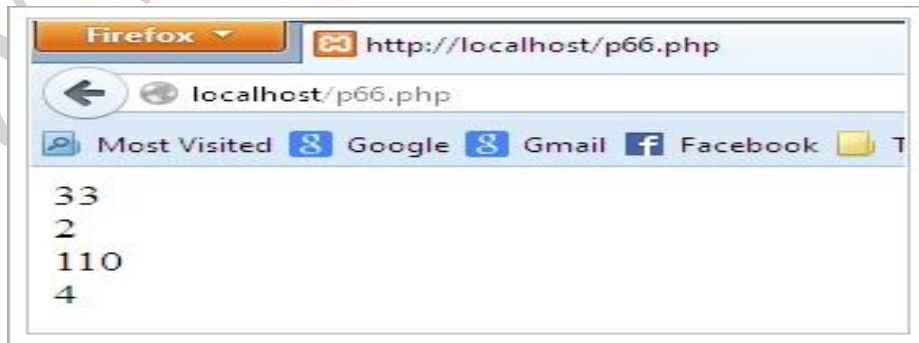
প্রোগ্রাম নং ৬৮ (p68.php) Arithmetic অপারেটর এর ব্যবহার

```
<?php  
$num1 = 4;  
$num2 = 5;  
$num3 = 2;  
$answer = $num1 * ($num2 - $num3);  
echo $answer;  
?>
```

উপরের ৬৮ নং প্রোগ্রামের : **Output: 12**

প্রোগ্রাম নং ৬৯ (p69.php) Arithmetic অপারেটর এর ব্যবহার।

```
<?php  
$a = 10;  
$b = 11;  
$c = 12;  
$add = $a + $b + $c;  
$sub = $c - $a;  
$mult = $a * $b;  
$div = $c / 3;  
echo $add, "<br> ", $sub, "<br>";  
echo $mult, "<br> ", $div, " <br>";  
echo "\n";  
?>
```



চিত্রঃ ৬.৫ প্রোগ্রাম p69 এর আউটপুট

প্রোগ্রাম নং ৭০ (p70.php) Arithmetic অপারেটর এর ব্যবহার

```
<?php
$value = 1;
echo $value++;

echo "<br />";
echo $value;
?>
```

উপরের ৭০ নং প্রোগ্রামের **Output: 1 2**

প্রোগ্রাম নং ৭১ (p71.php) Arithmetic অপারেটর এর ব্যবহার

```
<?php
// Increment
$counter = 15;
$counter ++; // 16
$counter ++; // 17
echo "Counter = " . $counter;
// Counter = 17
?>
```

উপরের ৭১ নং প্রোগ্রামের **Output: Counter = 17**

প্রোগ্রাম নং ৭২ (p72.php) Arithmetic অপারেটর এর ব্যবহার

```
<?php
$var1 = 3;
$var2 = 5;
$var1 = $var2; //var1=5
$var1 += $var2; //var1=8
$var1 -= $var2; //var1= -2
$var1 *= $var2; //var1=15
$var1 /= $var2; //var1=0.6
$var1 %= $var2; //var1=3
//you can print $var1 after each step to see its value
?>
```

৬.৬ অ্যাসাইনমেন্ট অপারেটরস (Assignment Operators) :

অ্যাসাইনমেন্ট অপারেটর ব্যবহার করা হয় কোন ভ্যারিয়েবল এর Value কে সেট করার জন্য। এজন্য equal to (=) সাইন ব্যবহার করা হয়। Assignment অপারেটর এর মাধ্যমে একটি ভ্যারিয়েবল এর Value অন্য ভ্যারিয়েবল এ সেট (Assign) করা যায়।



ভ্যালিউ ৪

\$m=4;



Assignment অপারেটর

ভ্যারিয়েবল m যার ভ্যালিউ 4

\$p=\$m;

এখানে p এর মান হচ্ছে m এর মান সুতরাং p এর মান হবে 4

\$s= "Bangladesh";

এখানে s এর Value হল Bangladesh

প্রোগ্রাম নং ৭৩ (p73.php) Assignment অপারেটর এর ব্যবহার

```
<?php
$a = "Hello ";
$b = $a . "World!"; // now $b contains "Hello World!"
$a = "Hello ";
$a .= "World!"; // now $a contains "Hello World!"
?>
```

প্রোগ্রাম নং ৭৪ (p74.php) Assignment অপারেটর এর ব্যবহার

```
<?php
class Dec {
    public function decrement() {
        $someVariable = "anything";
        $a = 3;
        $b = 4;
        $sum = $a + $b; // $sum now contains the value 7
    }
}
$p=new Dec;
$p-> decrement();
?>
```

+= প্লাস ইকুয়াল (Plus Equal)

$x+=y;$ \longrightarrow $x=x+y;$
 $\$a=3;$

অবজেক্ট অরিয়েটেড পিএইচপি

$\$a+=6;$ \longrightarrow $\$a=\$a+6;$

সুতরাং a এর মান হবে 9

$\$a+=6;$
 $\$a=\$a+6;$ } একই জিনিস দুইভাবে লিখা যায়
 $\$p+=2;$ \longleftrightarrow $\$p=\$p+2;$
 $\$x+=\$y;$ \longleftrightarrow $\$x=\$x+\$y;$

```
<?php
    $a=3;
    $a+=4;
    echo $a;
?>
```

প্রোগ্রাম নং ৭৫ (p75.php) Assignment অপারেটর এর ব্যবহার

```
<?php
    $a=3;
    $b=4;
    $c=$a+$b;
    $a+=$b;
    echo $a;
    echo $c;
?>
```

উপরের ৭৫ নং প্রোগ্রামের Output: 77

-= মাইনাস ইকুয়াল (Minus Equal)

$x=x-y;$
এর পরিবর্তে আমরা লিখতে পারি $x-=y;$
 $\$x=5;$
 $\$x-=2;$
এখানে $\$x$ মান হবে 3

প্রোগ্রাম নং ৭৬ (p76.php) Assignment অপারেটর এর ব্যবহার

```
<?php
    $a=8;
    $a-=3;
```

অবজেক্ট অরিয়েটেড পিএইচপি

```
echo $a;  
?>
```

উপরের ৭৬ নং প্রোগ্রামের Output: 5

প্রোগ্রাম নং ৭৭ (p77.php) Assignment অপারেটর এর ব্যবহার

```
<?php  
$a=12;  
$b=2;  
$c=$a-$b;  
$a=$b;  
echo $a;  
echo $c;  
?>
```

উপরের ৭৭ নং প্রোগ্রামের Output: 1010

*** = ইন্টু ইকুয়াল (Into Equal)**

$\$x*=\$y;$ \longleftrightarrow $\$x=\$x*\$y;$

$\$p=3;$

$\$p*=5;$

সুতরাং p এর মান হবে 15

কারণ $\$p*=5;$

$\$p=\$p*5;$

$=3*5;$

$=15;$

$\$m=6;$

$\$n=2;$

$\$m*=\$n;$ \longleftrightarrow $\$m=\$m*\$n;$

$=6*2;$

$=12;$

সুতরাং m এর মান 12

প্রোগ্রাম নং ৭৮ (p78.php) Assignment অপারেটর এর ব্যবহার

```
<?php  
$a=8;  
$a*=3;  
echo $a;  
?>
```

অবজেক্ট অরিয়েটেড পিএইচপি

উপরের ৭৮ নং প্রোগ্রামের : Output: 24

প্রোগ্রাম নং ৭৯ (p79.php) Assignment অপারেটর এর ব্যবহার

```
<?php
    $a=12;
    $b=2;
    $c=$a*$b;
    $a*=$b;
    echo $a;
    echo $c;
?>
```

উপরের ৭৯ নং প্রোগ্রামের : Output: 24, 24

/= ডিভিশন ইকুয়াল (Division Equal)

```
$h/=$m; ➡ $h=$h/$m;
$n=12;
$n/=3; ➡ $n=$n/3;
           =12/3;
           =4;
```

n এর মান হবে 4

```
$p=24;
```

```
$q=3;
```

```
$p/=$q;
```

p এর মান হবে 8

প্রোগ্রাম নং ৮০ (p80.php) Assignment অপারেটর এর ব্যবহার

```
<?php
    $a=8;
    $a/=2;
    echo $a;
?>
```

উপরের ৮০ নং প্রোগ্রামের : Output: 4

প্রোগ্রাম নং ৮১ (p81.php) Assignment অপারেটর এর ব্যবহার

```
<?php
    $a=12;
    $b=2;
    $c=$a/$b;
```

অবজেক্ট অরিয়েটেড পিএইচপি

```
$a=$b;  
echo $a;  
echo $c;
```

?>

উপরের চ-১ নং প্রোগ্রামের Output: 66

.= কনক্যাটেনেশন ইকুয়াল (Concatenation Equal)

```
$x.$y;     ⇒     $x=$x.$y;  
$x=6;  
$x.=8;     ⇒     $x=$x.8;  
              = 6.8  
              = 68
```

সুতরাং x এর মান হবে 68

```
$p=2;  
$m= 'a';  
$c=$p.$m;  
  =2.a;  
  =2a;
```

সুতরাং c এর মান হবে 2a

প্রোগ্রাম নং চ-২ (p82.php) Assignment অপারেটর এর ব্যবহার

```
<?php  
  $a=8;  
  $a.=2;  
  echo $a;
```

?>

উপরের চ-২ নং প্রোগ্রামের Output: 82

প্রোগ্রাম নং চ-৩(p83.php) Assignment অপারেটর এর ব্যবহার

```
<?php  
  $a=12;  
  $b=2;  
  $c=$a.$b;  
  $a.=$b;  
  echo $a;  
  echo $c;
```

?>

উপরের চ-৩ নং প্রোগ্রামের Output: 122122

প্রোগ্রাম নং ৮৪ (p84.php) Assignment অপারেটর এর ব্যবহার

```
<?php
    $a= "Hello";
    $a.= "world ";
    echo $a;
?>
```

উপরের ৮৪ নং প্রোগ্রামের **Output:** Helloworld

%= মডিউলাস ইকুয়াল টু (Modulus Equal to)

```
$x%=$y; ➡ $x=$x%$y;
$x=18;
$x%=4; ➡ $x%=4;
           $x=$x%4;
           =18%4;
           =2;
```

এখানে x এর মান হবে 2 কারণ আমরা জানি, মডিউলাস সব সময় ভাগশেষ দেয়। সেজন্য 18 কে 4 দিয়ে ভাগ করলে ভাগশেষ হবে 2

```
$m=14;
$n=3;
$m%=$n;
m এর মান হবে 2;
$x=17;
$y=5;
$x%=$y;
$x=$x%$y;
  =17%5;
  =2;
```

প্রোগ্রাম নং ৮৫ (p85.php) Assignment অপারেটর এর ব্যবহার

```
<?php
    $a=9;
    $a%=2;
    echo $a;
?>
```

অবজেক্ট অরিয়েটেড পিএইচপি

উপরের ৮৫ নং প্রোগ্রামের Output: 1

প্রোগ্রাম নং ৮৬(p86.php) Assignment অপারেটর এর ব্যবহার

```
<?php
    $a=12;
    $b=3;
    $c=$a%$b;
    $a%=$b;
    echo $a;
    echo $c;
?>
```

উপরের ৮৬ নং প্রোগ্রামের Output: 0 0

অ্যারিথমেটিক অপারেটরস (Assignment Operators)

Operator	Example	Is The Same As
=	x=y	x=y
+=	x+=y	x=x+y
-=	x-=y	x=x-y
=	x=y	x=x*y
/=	x/=y	x=x/y
.=	x.=y	x=x.y
%=	x%=y	x=x%y

প্রোগ্রাম নং ৮৭ (p87.php) Assignment অপারেটর এর ব্যবহার

```
<?php
$a = 3;
$a += 5; // sets $a to 8, as if we had said: $a = $a + 5;
$b = "Hello ";
$b .= "There!"; // sets $b to "Hello There!", just like $b = $b . "There!";
?>
```

প্রোগ্রাম নং ৮৮ (p88.php) Assignment অপারেটর এর ব্যবহার

```
<?php
/*This Block of PHP Code represent the
use of PHP ASSIGNMENT OPERATORS*/
 $x=5; //Variable 'x' with value 5.
 $y=3; //Variable 'y' with value 3.
 // Use of += operator.
 $x+= $y;
 echo 'Result Of Addition: ';
 echo $x;
 echo '<br/>';
 // Use of -= operator.
 $x-= $y;
 echo 'Result Of Subtraction: ';
 echo $x;
 echo '<br/>';

 // Use of *= operator.
 $x*= $y;
 echo 'Result Of Multiplication: ';
 echo $x;
 echo '<br/>';
 // Use of /= operator.
 $x/= $y;
 echo 'Result Of Division: ';
 echo $x;
 echo '<br/>';
 // Use of .= operator.
 $x.= $y;
 echo 'Result Of Concatenation: ';
 echo $x;
 echo '<br/>';
 // Use of %= operator.
 $x%= $y;
 echo 'Result Of Modulus: ';
 echo $x;
 echo '<br/>';
 // Use of = operator.
 $x= $y;
 echo 'Result Of Assignment: ';
 echo $x;
 echo '<br/>';
?>
```



চিত্রঃ ৬.৬ প্রোগ্রাম p88 এর আউটপুট

প্রোগ্রাম নং ৮৯ (p89.php) Assignment অপারেটর এর ব্যবহার

```
<?php
$a = 3;
$a += 5; // sets $a to 8, as if we had said: $a = $a + 5;
$b = "Hello ";
$b .= "There!"; // sets $b to "Hello There!", just like $b = $b . "There!";
?>
```

প্রোগ্রাম নং ৯০ (p90.php) Assignment অপারেটর এর ব্যবহার

```
<html>
  <body>
    <?php
      $initial_value = 10;
      $initial_value +=;
    ?>
  </body>
</html>
```

প্রোগ্রাম নং ৯১ (p91.php) Assignment অপারেটর এর ব্যবহার

```
<html>
```

```
<body>
  <?php
    $example_variable = "An example";
    $number_variable = 1;
  ?>
</body>
</html>
```

প্রোগ্রাম নং ৯২ (p92.php) Assignment অপারেটর এর ব্যবহার

```
<?php
$my_var = 12;
$my_var += 14; // $my_var now equals 26
$my_var -= 12; // $my_var now equals 14
$my_var *= 10; // $my_var now equals 140
$my_var /= 7; // $my_var now equals 20
$my_var %= 6; // $my_var now equals 2
?>
```

প্রোগ্রাম নং ৯৩ (p93.php) Assignment অপারেটর এর ব্যবহার

```
<?php
$x = 45;
$x--;
echo $x;
$x -= 1;
echo $x;
$x = $x - 1;
echo $x;
?>
```

উপরের ৯৩ নং প্রোগ্রামের : **Output:** 444342

প্রোগ্রাম নং ৯৪ (p94.php) Assignment অপারেটর এর ব্যবহার

```
<?php
$x = 15;
$x++;
echo $x;
$x += 1;
```

অবজেক্ট অরিয়েটেড পিএইচপি

```
echo $x;  
$x = $x + 1;  
echo $x;
```

?>

উপরের ৯৪ নং প্রোগ্রামের Output: 161718

প্রোগ্রাম নং ৯৫ (p95.php) Assignment অপারেটর এর ব্যবহার

```
<?php  
$counter = 8;  
$counter += 1;  
echo $counter;
```

?>

উপরের ৯৫ নং প্রোগ্রামের Output: 9

প্রোগ্রাম নং ৯৬ (p96.php) Assignment অপারেটর এর ব্যবহার

```
<?php  
$counter = 8;  
$counter = $counter + 1;  
echo $counter;
```

?>

উপরের ৯৬ নং প্রোগ্রামের Output: 9

প্রোগ্রাম নং ৯৭ (p97.php) Assignment অপারেটর এর ব্যবহার

```
<?php  
$a=($b=23)+12;  
echo "Value of a is :".$a."<br/>";  
$a+=34;  
echo "Value of a is :".$a."<br/>";  
$a-=34;  
echo "Value of a is :".$a."<br/>";  
$a/=34;  
echo "Value of a is :".$a."<br/>";  
echo "Value in integer data type of a is :".(int)$a."<br/>";  
$a*=34;  
echo "Value of a is :".$a."<br/>";  
$a%=34;  
echo "Value of a is :".$a."<br/>";  
echo "Value of b is :".$b;
```

?>

```
Value of a is :35
Value of a is :69
Value of a is :35
Value of a is :1.02941176471
Value in integer data type of a is :1
Value of a is :35
Value of a is :1
Value of b is :23
```

চিত্রঃ ৬.৬ প্রোগ্রাম p97 এর আউটপুট

প্রোগ্রাম নং ৯৮ (p98.php) Assignment অপারেটর এর ব্যবহার

```
<?php
$a = 3;
$a += 5; // sets $a to 8, as if we had said: $a = $a + 5;
$b = "Hello ";
$b .= "There!"; // sets $b to "Hello There!", just like $b = $b "There!";
?>
```

প্রোগ্রাম নং ৯৯ (p99.php) Assignment অপারেটর এর ব্যবহার

```
<?php
$a = 'a';
$b = 'b';
$a .= $b .= "foo";
echo $a, "\n", $b;
?>
```

উপরের ৯৯ নং প্রোগ্রামের Output: abfoo bfoo

প্রোগ্রাম নং ১০০ (p100.php) Assignment অপারেটর এর ব্যবহার

```
<?php
$a = 3;
$a += 5; // sets $a to 8, as if we had said: $a = $a + 5;
$b = "Hello ";
$b .= "There!"; // sets $b to "Hello There!", just like $b = $b . "There!";
?>
```

- $A += B$ is the same as $(A = A + B)$
- $A -= B$ is the same as $(A = A - B)$
- $A *= B$ is the same as $(A = A * B)$

- $A /= B$ is the same as $(A = A / B)$
- $A \% = B$ is the same as $(A = A \% B)$
- $A \& = B$ is the same as $(A = A \& B)$
- $A |= B$ is the same as $(A = A | B)$
- $A ^= B$ is the same as $(A = A ^ B)$
- $A << = B$ is the same as $(A = A << B)$
- $A >> = B$ is the same as $(A = A >> B)$

c-109 Assignment অপারেটরস।

প্রথাম নং ১০১ (p101.php) Assignment অপারেটর এর ব্যবহার

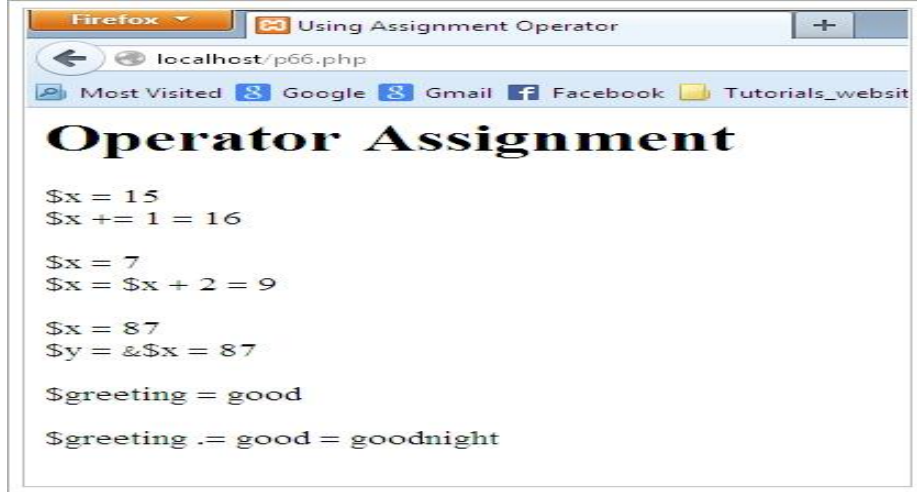
```
<html>
<head><title>Assignment Operators</title></head>
<body>
  <?php
    $a = 42;
    $b = 20;
    $c = $a + $b; /* Assignment operator */
    echo "Addition Operation Result: $c <br/>";
    $c += $a; /* c value was 42 + 20 = 62 */
    echo "Addition AND Assignment Operation Result: $c <br/>";
    $c -= $a; /* c value was 62 - 20 = 42 */
    echo "Subtract AND Assignment Operation Result: $c <br/>";
    $c *= $a; /* c value was 42 * 20 = 840 */
    echo "Multiply AND Assignment Operation Result: $c <br/>";
    $c /= $a; /* c value was 840 / 20 = 42 */
    echo "Division AND Assignment Operation Result: $c <br/>";
    $c %= $a; /* c value was 42 % 20 = 2 */
    echo "Modulus AND Assignment Operation Result: $c <br/>";
  ?>
</body>
</html>
```




চিত্রঃ ৬.৬ প্রোগ্রাম p101 এর আউটপুট

প্রোগ্রাম নং ১০২ (p102.php) Assignment অপারেটর এর ব্যবহার

```
<html>
  <head><title>Using Assignment Operator</title> </head>
<body>
  <h1>Operator Assignment</h1>
  <?
    $x = 15;
    echo ("\$x = $x <br>");
    $x += 1 ;
    echo ("\$x += 1 = $x <br><br>");
    $x = 7;
    echo ("\$x = $x <br>");
    $x = $x + 2;
    echo ("\$x = \$x + 2 = $x <br><br>");
    $x = 87;
    echo ("\$x = $x <br>");
    $y = &$x;
    echo ("\$y = &\$x = $x <br><br>");
    $greeting = "good";
    echo ("\$greeting = $greeting <br><br>");
    $greeting .= "night";
    echo ("\$greeting .= good = $greeting");
  ?>
</body> </html>
```

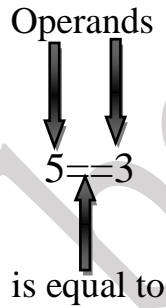


চিত্রঃ ৬.৬ শোখাম p102 এর আউটপুট

৬.৭ কম্প্যারিজন অপারেটরস (Comparison Operators) :

= ইজ ইকুয়াল টু (is equal to)

== ইজ ইকুয়াল টু (পর পর দুটি "=" symbol এবং এর দু'পাশে দুটি অপারেন্ড অবশ্যই থাকতে হবে।)



উপরের Comparison এর রেজাল্ট মিথ্যা হবে, কারণ 5 এবং 3 সমান নয়।

== অপারেটর টি চেক করে তার দু'পাশের দুটি অপারেন্ড সমান কি না?

\$a=6

\$a==5 ➡ এটি মিথ্যা (False)

\$b=6

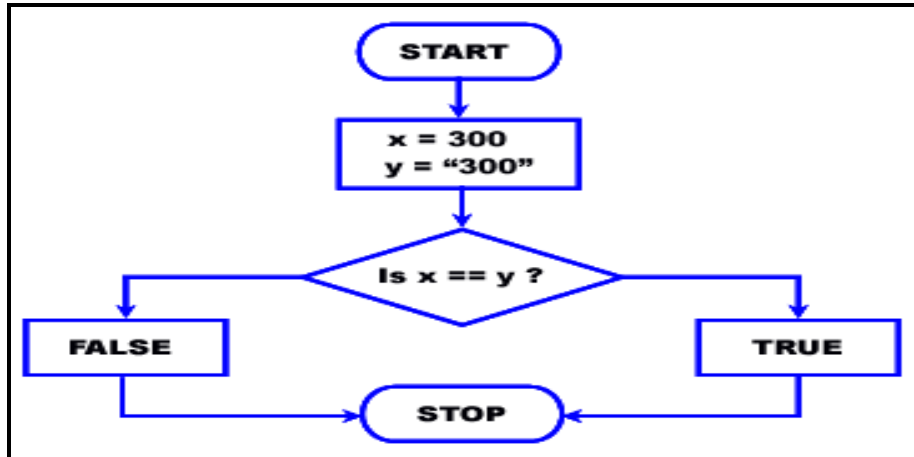
\$a==\$b ➡ এটি সত্য (True)

কারণ a এর মান 6 এবং b এর মানও 6. এজন্য দুটি সমান হয়েছে এবং কন্ডিশনটি সত্য হয়েছে।

\$x= 9

\$y= 2

\$x==\$y এটি মিথ্যা



চিত্র 6.7 ইজ ইকুয়াল টু অপারেটর এর ফ্লো-চার্ট

```
<?php  
$x = 300;  
$y = "300";  
var_dump($x === $y);  
?>
```



চিত্রঃ উপরের প্রোগ্রামের আউটপুট

প্রোগ্রাম নং ১০৩ (p103.php) Comparison অপারেটর এর ব্যবহার

```
<?php  
$x = 30;  
$y = 25;  
var_dump($x === $y);  
?>
```

উপরের ১০৩ নং প্রোগ্রামের Output: bool(false)


প্রোগ্রাম নং ১০৪ (p104.php) Comparison অপারেটর এর ব্যবহার


```
<?php
$x = 3; $y = 3;
var_dump($x === $y);
?>
```

```
<?php
Class Com {
    public function comparison() {
        $x = 3;
        $y = 3;
        var_dump($x === $y);
    }
}
$p=new Com;
$p-> comparison();
?>
```

!= ইজ নট ইকুয়াল (is not equal)

অর্থাৎ এটি দ্বারা বুঝায়, যদি এই অপারেটর এর দুই পাশের দুটি অপারেন্ড সত্য (সমান/একই রকম) না হয়, তাহলে এটি সত্য (True) হবে। আর সত্য (সমান/একই রকম) হলে, এর মান মিথ্যা (False) হবে।

6!=3  এটি সত্য কারণ, 6 এবং 3 সমান নয়।

6!=6  এটি মিথ্যা কারণ, 6 এবং 6 সমান।

\$a=5

\$b=9

\$a!=9  সত্য

\$a!=5  মিথ্যা

\$a!=\$b  সত্য

\$b!=\$a  সত্য

9!=\$a  সত্য

```
<?php
$x = 300;
$y = 300;
var_dump($x != $y);
?>
```

প্রোগ্রাম নং ১০৫ (p105.php) Comparison অপারেটর এর ব্যবহার

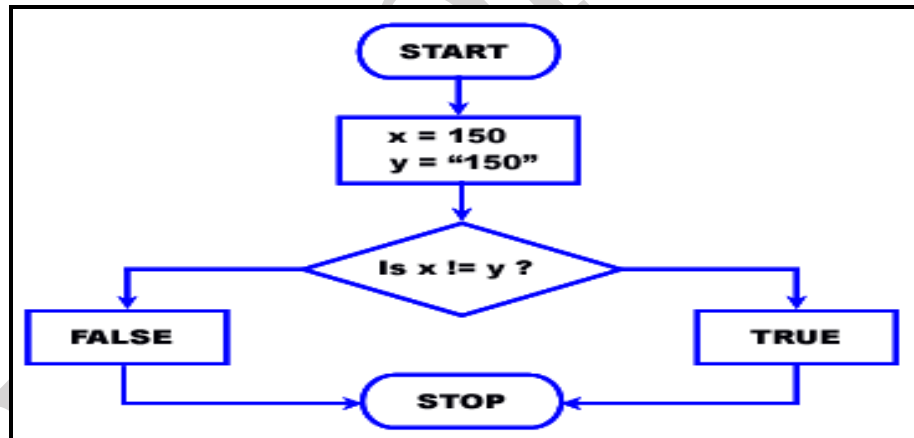
```
<?php
    $x = 30;
    $y = 300;
    var_dump($x != $y);
?>
```

উপরের ১০৫ নং প্রোগ্রামের Output: bool(true)

প্রোগ্রাম নং ১০৬ (p106.php) Comparison অপারেটর এর ব্যবহার

```
<?php
    $x = 50;
    $y = 50;
    var_dump($y != $x);
?>
```

উপরের ১০৬ নং প্রোগ্রামের Output: bool(false)



চিত্র 6.7 ইজ নট ইকুয়াল অপারেটর এর ফ্লো-চার্ট

প্রোগ্রাম নং ১০৭ (p107.php) Comparison অপারেটর এর ব্যবহার

```
<?php
    $x = 150;
    $y = "150";
    var_dump($x != $y);
?>
```

উপরের ১০৭ নং প্রোগ্রামের Output: bool(false)

> ইজ গ্রেটার দেন (is greater than)

এই অপারেটরটি চেক করে ডানের অপারেণ্ডটি বামের চেয়ে বড় কি না? যদি বড় হয়, তাহলে সত্যি হয়। অন্যথায় মিথ্যা।

`var1 > var2`

`var1` অবশ্যই `var2` এর চেয়ে বড় হতে হবে, তাহলে কন্ডিশনটি সত্যি হবে।

`5 > 2` \longrightarrow এটি সত্যি কারণ, 2 হতে 5 বড়।

`2 > 5` \longrightarrow এটি মিথ্যা (False) কারণ 2,5 এর চেয়ে ছোট।

`$a > $b`

যদি `$a` বড় হয় তাহলে সত্যি হবে, আর `$a` ছোট হলে মিথ্যা হবে।

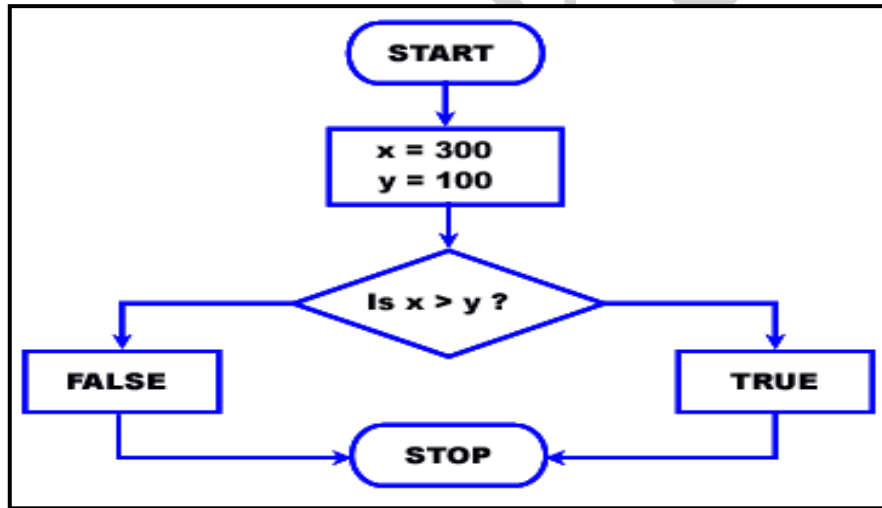
`$a = 6`

`$b = 4`

`$a > $b` \longrightarrow সত্যি

`$a > 10` \longrightarrow মিথ্যা

`5 > $a` \longrightarrow মিথ্যা



চিত্র ৬.৭ : ইজ গ্রেটার দেন অপারেটর এর ফ্লো-চার্ট।

```
<?php  
$x = 200;  
$y = 300;  
var_dump($x > $y);  
?>
```

উপরের প্রোগ্রামের : **Output:** bool(false)

প্রোগ্রাম নং ১০৮ (p108.php) Comparison অপারেটর এর ব্যবহার

```
<?php
    $x = 30;
    $y = 300;
    var_dump($x > $y);
?>
```

উপরের ১০৮ নং প্রোগ্রামের Output: bool(false)

প্রোগ্রাম নং ১০৯ (p109.php) Comparison অপারেটর এর ব্যবহার

```
<?php
    $x = 50;
    $y = 10;
    var_dump($y > $x);
?>
```

উপরের ১০৯নং প্রোগ্রামের Output: bool(false)

< ইজ লেজ দেন (is less than)

এটি greater than এর উল্টো।

$var1 < var2$

var1, var2 এর চেয়ে ছোট হবে, তাহলেই সত্য হবে। অন্যথায় মিথ্যা হবে।

$6 < 8$ \Rightarrow এটি সত্য কারণ 6,8 এর চেয়ে ছোট।

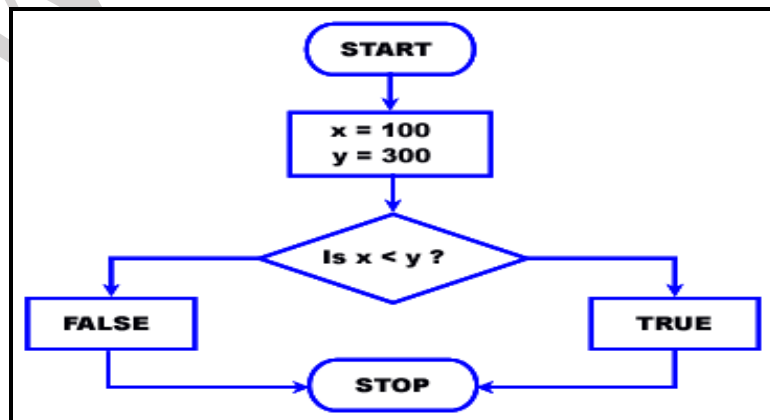
$\$a = 9$

$\$b = 6$

$\$b < \a \Rightarrow সত্য

$\$a < \b \Rightarrow মিথ্যা

$\$a < 15$ \Rightarrow সত্য



চিত্র ৬.৭ : ইজ লেজ দেন অপারেটর এর ফ্লো-চার্ট

```
<?php
    $x = 100;
    $y = 300;
    var_dump($x<$y);
?>
```

উপরের প্রোগ্রামের **Output:** bool(true)

প্রোগ্রাম নং ১১০ (p110.php) Comparison অপারেটর এর ব্যবহার

```
<?php
    $x = 50;
    $y = 10;
    var_dump($y < $x);
?>
```

উপরের ১১০নং প্রোগ্রামের **Output:** bool(true)

প্রোগ্রাম নং ১১১ (p111.php) Comparison অপারেটর এর ব্যবহার

```
<?php
    $x = 45;
    $y = 95;
    var_dump($y > $x);
?>
```

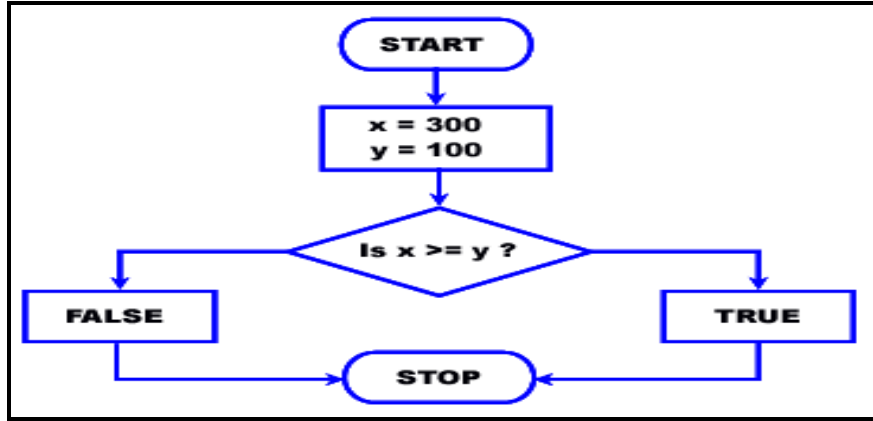
উপরের ১১১নং প্রোগ্রামের **Output:** bool (true)

>= ইজ হেটার দেন অর ইকুয়াল টু (is greater than or equal to)

var1>=var2

var1 যদি var2 এর চেয়ে বড় অথবা সমান হয় তাহলে সত্যি হবে। অন্যথায় মিথ্যা হবে।

5>=2	➡	সত্যি
6>=8	➡	মিথ্যা
9>=9	➡	সত্যি
\$a=9		
\$b=6		
\$a>=9	➡	সত্যি
\$a>=\$b	➡	মিথ্যা



চিত্র : ৬.৭ ইজ ছোটর দেন অর ইকুয়াল টু অপারেটর এর ফ্লো-চার্ট

```
<?php  
$x = 300;  
$y = 100;  
var_dump($x>=$y);  
?>
```

উপরের প্রোগ্রামের : **Output:** bool(true)

প্রোগ্রাম নং ১১২ (p112.php) Comparison অপারেটর এর ব্যবহার

```
<?php  
$x = 50;  
$y = 10;  
var_dump($y >= $x);  
?>
```

উপরের ১১২নং প্রোগ্রামের : **Output:** bool(false)

প্রোগ্রাম নং ১১৩ (p113.php) Comparison অপারেটর এর ব্যবহার

```
<?php  
$x = 45;  
$y = 95;  
var_dump($y >= $x);  
?>
```

উপরের ১১৩নং প্রোগ্রামের : **Output:** bool(true)

<= ইজ লেজ দেন অর ইকুয়াল টু (is less than or equal to)

```
var1<=var2
```

var1 যদি var2 এর চেয়ে ছোট অথবা সমান হয় তাহলে সত্যি হবে। অন্যথায় মিথ্যা হবে।

$5 \leq 8$ এটি সত্যি কারণ, ডানের অপারেভ বামের চেয়ে ছোট।

$\$x \leq \y

x যদি y এর চেয়ে ছোট হয় তাহলে সত্যি হবে।

x যদি y এর চেয়ে বড় হয় তাহলে মিথ্যা হবে।

$\$x = 5$

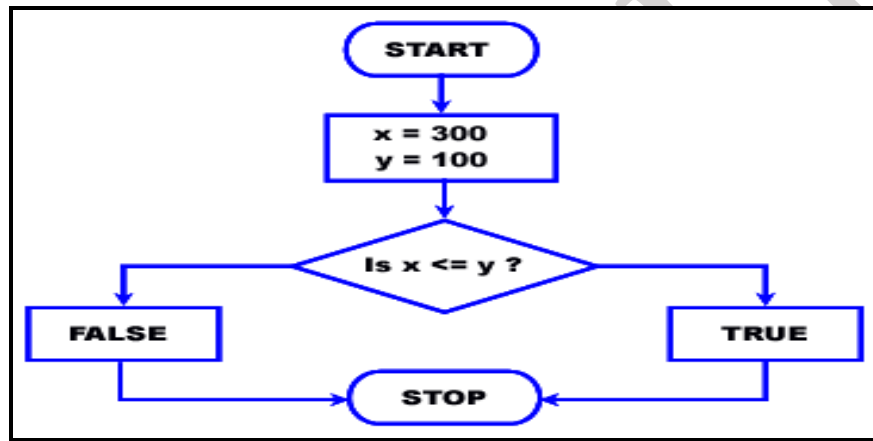
$\$y = 8$

$\$x \leq \y \Rightarrow সত্যি

$\$x \leq 50$ \Rightarrow সত্যি

$\$y \leq \x \Rightarrow মিথ্যা

$\$x \leq 5$ \Rightarrow সত্যি



চিত্র : ৬.৭ ইজ লেজ দেন অর ইকুয়াল টু অপারেটর এর ফ্লো-চার্ট

```
<?php  
$x = 300;  
$y = 100;  
var_dump($x <= $y);
```

?>

উপরের প্রোগ্রামের : **Output: bool(false)**

প্রোগ্রাম নং ১১৪ (p114.php) Comparison অপারেটর এর ব্যবহার

```
<?php  
$x = 50;  
$y = 10;  
var_dump($y <= $x);
```

?>

উপরের ১১৪নং প্রোগ্রামের : **Output: bool(true)**

প্রোগ্রাম নং ১১৫ (p115.php) Comparison অপারেটর এর ব্যবহার

```
<?php
    $x = 45;
    $y = 95;
    var_dump($y <= $x);
?>
```

উপরের ১১৫ নং প্রোগ্রামের Output: bool (false)

কমপ্যারিজন অপারেটরস (Comparison Operators)

Operator	Description	Example
==	is equal to	5==8 returns false
!=	is not equal	5!=8 returns true
>	is greater than	5>8 returns false
<	is less than	5<8 returns true
>=	is greater than or equal to	5>=8 returns false
<=	is less than or equal to	5<=8 returns true

c-131

```
<?php
    $num1 = 9;
    $num2 = 9;
    $num3 = 22;
?>
```

প্রোগ্রাম নং ১১৬ (p116.php)

```
<?php
    //returns true if the variable on left is not equal to right..
    $res = ($num1 <> $num2);
    //returns false..
?>
```

প্রোগ্রাম নং ১১৭ (p117.php) Comparison অপারেটর এর ব্যবহার

```
<?php
    //returns true if both variables are equal to each other
```

```
$res = ($num1 == $num2);  
//returns true..  
?>
```

প্রোগ্রাম নং ১১৮ (p118.php) Comparison অপারেটর এর ব্যবহার

```
<?php  
//returns true if variable on left is less than or equal to right..  
$res = ($num3 <= $num1);  
//returns false  
?>
```

প্রোগ্রাম নং ১১৯ (p119.php) Comparison অপারেটর এর ব্যবহার

```
<?php  
//Returns true if the variable in left side is greater that the right..  
$res = ($num3 > $num1);  
//returns true..  
?>
```

প্রোগ্রাম নং ১২০ (p120.php) Comparison অপারেটর এর ব্যবহার

```
<?php  
$a = 3; $b = 3; $c = 10;  
// all of the following are true  
echo ($a == $b);  
echo ($a != $c);  
echo (8 === 8);  
echo ("8" !== 8);  
echo ($b < $c);  
echo ($b <= $b);  
echo ($c > $b);  
echo ($c >= $c);  
?>
```

উপরের ১২০ নং প্রোগ্রামের Output: 11111111

প্রোগ্রাম নং ১২১ (p121.php) Comparison অপারেটর এর ব্যবহার

```
<? php  
$a=12;  
$b=12;  
var_dump ($a==$b);echo "<br/>";  
var_dump ($a=== $b);echo "<br/>";  
var_dump ($a!= $b);echo "<br/>";
```

অবজেক্ট অরিয়েটেড পিএইচপি

```
var_dump ($a<>$b);echo "<br/>";  
var_dump ($a<$b);echo "<br/>";  
var_dump ($a>$b);echo "<br/>";  
var_dump ($a<=$b);echo "<br/>";  
var_dump ($a>=$b);echo "<br/>";  
?>
```

উপরের ১২১নং প্রোগ্রামের Output:

```
bool(true) bool(true)  
bool(false) bool(false)  
bool(false) bool(false)  
bool(true) bool(true)
```

প্রোগ্রাম নং ১২২ (p122.php) Comparison অপারেটর এর ব্যবহার

```
<?php  
var_dump("23"==23);  
echo "<br/>";  
var_dump("23"===23);  
?>
```

A screenshot of a PHP script's output. The text is displayed in a monospaced font within a rectangular box. The first line shows 'bool(true)' and the second line shows 'bool(false)'. The text is slightly blurred, suggesting it was captured from a screen.

চিত্রঃ ৬.৭ প্রোগ্রাম p122 এর আউটপুট

প্রোগ্রাম নং ১২৩ (p123.php) Comparison অপারেটর এর ব্যবহার

```
<?php  
$a=23;  
$b=45;  
$c=($a>$b)?$a:$b;  
echo "Larger number is= ".$c; ?>
```

উপরের ১২৩নং প্রোগ্রামের Output: Larger number is= 45

প্রোগ্রাম নং ১২৪ (p124.php) Comparison অপারেটর এর ব্যবহার

```
<?php  
$var1 = 5;  
$var2 = 3;
```

```
// Is Greater Than Comparison  
echo $var1 > $var2; // should output a "1" meaning TRUE ?>
```

উপরের ১২৪নং প্রোগ্রামের : **Output:** Larger number is= 1

প্রোগ্রাম নং ১২৫ (p125.php) Comparison অপারেটর এর ব্যবহার

```
<?php  
var_dump ("23"==23);  
echo "<br/>";  
var_dump ("23"===23);
```

উপরের ১২৫নং প্রোগ্রামের : **Output:**

```
bool(true)  
bool(false)
```

প্রোগ্রাম নং 126 (p126.php) Comparison অপারেটর এর ব্যবহার

```
<?php  
// define some variables  
$mean = 29;  
$median = 40;  
$mode = 29;  
  
// less-than operator  
// return true if left side is less than right  
// return true here  
$result = ($mean < $median);  
// greater -than operator  
// returns true if left side is greater than right  
// return false here  
$result = ($mean > $median);  
  
// less than or equal to operator  
// return true if left side is less than or equal to right  
// return false here  
$result =($median <= $mode);  
// greater than or equal to operator  
// return true if left side is greater than or equal to right  
// return true here  
$result = ($median >= $mode);  
// equality operator  
// return true if left side is equal to right  
// return true here
```

```
$result = ($mean== $mode);  
// not equal to operator  
// return true if left side is not equal to right  
// return true here  
$result = ($mean != $mode);  
// inequality operator  
// return true if left side is not equal to right  
// return true here  
$result = ($mean <> $mode);  
?>
```

প্রোগ্রাম নং ১২৭ (p127.php) Comparison অপারেটর এর ব্যবহার

```
<? php  
$a=12;  
$b=12;  
var_dump ($a==$b);echo "<br/>";  
var_dump ($a=== $b);echo "<br/>";  
var_dump ($a!= $b);echo "<br/>";  
var_dump ($a<> $b);echo "<br/>";  
var_dump ($a< $b);echo "<br/>";  
var_dump ($a> $b);echo "<br/>";  
var_dump ($a<= $b);echo "<br/>";  
var_dump ($a>= $b);echo "<br/>";  
?>
```

উপরের ১২৭নং প্রোগ্রামের Output:

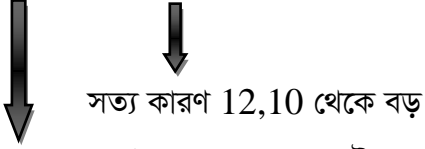
```
bool(true) bool(true)  
bool(false) bool(false)  
bool(false) bool(false)  
bool(true) bool(true)
```

৬.৮ লজিক্যাল অপারেটর (Logical Operators) :&& (and)

এই অপারেটর টিকে AND অথবা and এভাবে ও লিখা যায়।
\$variable1 && \$variable2

যদি \$variable1 এবং \$variable2 কন্ডিশন দুটি সত্য হয়, তাহলে উপরের এক্সপ্রেশনটি সত্য হবে। অন্যথায় মিথ্যা হবে।

$5 < 6 \ \&\& \ 12 > 10$



সত্য কারণ 5,6 থেকে থেকে ছোট

যেহেতু দুটো কন্ডিশনই সত্য, সেজন্য পুরো এক্সপ্রেশনটি সত্য হবে (True/1)।

সত্য \Rightarrow True \Rightarrow 1

মিথ্যা \Rightarrow False \Rightarrow 0

$\$a=6; \$c=15;$

$\$b=4;$

$\$a > \$b \ \&\& \ \$c > \$b; \Rightarrow$ সত্য

$\$a > 2 \ \&\& \ \$c > 2; \Rightarrow$ সত্য

$15 < 8 \ \&\& \ 25 > 9 \Rightarrow$ মিথ্যা

এখানে, প্রথম কন্ডিশনটি মিথ্যা কারণ, 15,8 থেকে ছোট না সেজন্য এটি মিথ্যা এবং দ্বিতীয় কন্ডিশনটি সত্য।

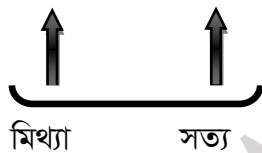
যেহেতু, উভয় কন্ডিশন সত্য না, সেজন্য এটি মিথ্যা হয়ে গেল।

$\$x=6;$

$\$y=15;$

$\$z=2;$

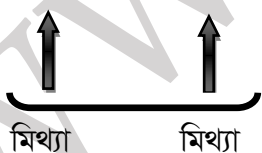
$\$x < \$z \ \&\& \ 2 < 15;$



মিথ্যা

মিথ্যা

$\$z > \$y \ \&\& \ \$y < \x



মিথ্যা

মিথ্যা

অর্থাৎ && (and) অপারেটর এর ক্ষেত্রে যেকোন একটি মিথ্যা হলেই মিথ্যা।

দুটো যদি সত্য হয়, তাহলে সত্য।

&& AND এর ক্ষেত্রে:

সত্য	AND	সত্য	→	সত্য
মিথ্যা	AND	মিথ্যা	→	মিথ্যা
সত্য	AND	মিথ্যা		মিথ্যা
মিথ্যা	AND	সত্য		মিথ্যা

```
<?php
    $x = 300;
    $y = 100;
    var_dump($x<$y && 5>20);
?>
```

উপরের প্রোগ্রামের **Output:** bool(false)

প্রোগ্রাম নং ১২৮ (p128.php) Logical অপারেটর এর ব্যবহার

```
<?php
    $x = 50;
    $y = 10;
    var_dump($y <= $x && $x > $y);
?>
```

উপরের ১২৮নং প্রোগ্রামের **Output:** bool(true)

প্রোগ্রাম নং ১২৯ (p129.php) Logical অপারেটর এর ব্যবহার

```
<?php
    $x = 45;
    $y = 95;
    $p=5 ;
    $q= 10 ;
    var_dump($y > $x && $p< $q);
?>
```

উপরের ১২৯নং প্রোগ্রামের **Output:** bool(true)

|| অর (or)

(or) এভাবেও লিখা যায়।

condition1 || condition2

কন্ডিশন ১ এবং ২ এর যে কোন একটি অথবা দুটি সত্যি হলে পুরো কন্ডিশন সত্য। অন্যথায় মিথ্যা হবে।

$5==6 \parallel 5<6$ এটি সত্য কারণ প্রথম কন্ডিশন মিথ্যা, কিন্তু দ্বিতীয় কন্ডিশনটি সত্যি সেজন্য সত্যি হবে।

$\$x=6; \$y=5;$

$\$k=8;$

$\$x>\$y \parallel \$k>\$y; \implies$ সত্য

$\$x==6 \parallel \$k==5; \implies$ সত্য

$\$y==8 \parallel \$x<15;$



সত্য

$\$k==8 \parallel \$k==9$



সত্য

$\$k<\$y \parallel \$x==\y



মিথ্যা
মিথ্যা
মিথ্যা

or এর ক্ষেত্রে দুটোই মিথ্যা হলে মিথ্যা অন্যথায় সব সময় সত্য।

|| OR/or এর ক্ষেত্রে:

সত্য OR সত্য \implies সত্য

মিথ্যা OR মিথ্যা \implies মিথ্যা

সত্য OR মিথ্যা \implies সত্য

মিথ্যা OR সত্য \implies সত্য

```
<?php  
$x = 300;
```

অবজেক্ট অরিয়েটেড পিএইচপি

```
$y = 100;  
var_dump($x<$y || 5>20);  
?>
```

প্রোগ্রাম নং ১৩০ (p130.php) Logical অপারেটর এর ব্যবহার

```
<?php  
$x = 50;  
$y = 10;  
var_dump($y <= $x || $x>$y);  
?>
```

উপরের ১৩০নং প্রোগ্রামের : **Output:** bool(true)

প্রোগ্রাম নং ১৩১ (p131.php): Logical অপারেটর এর ব্যবহার

```
<?php  
$x = 45;  
$y = 95;  
$p=5 ;  
$q= 10 ;  
var_dump($y > $x || $p< $q);  
?>
```

উপরের ১৩১নং প্রোগ্রামের : **Output:** bool(true)

প্রোগ্রাম নং ১৩২ (p132.php) Logical অপারেটর এর ব্যবহার

```
<?php  
$x = 50;  
$y = 10;  
var_dump($y < $x || $x==50);  
?>
```

উপরের ১৩২নং প্রোগ্রামের : **Output:** bool(true)

! নট (not)

এটিকে NOT (not) এভাবেও লিখা যায়।

!(5==5) \longrightarrow এটি মিথ্যা

!(\$v1==\$v2)

এটি \$v1 এবং \$v2 মিলে যা হবে তাকে উল্টে দিবে।

!(2= =2) \longleftarrow এখানে 2 এবং 2 সমান তাই ভিতরে সত্য হবে, কিন্তু, যেহেতু এখানে !(NOT) আছে, তাই এটি মিথ্যা হবে।

অবজেক্ট অরিয়েটেড পিএইচপি

!(3= =3) \Rightarrow মিথ্যা

!(3= =9) \Rightarrow সত্য

শেষের এক্সপ্রেশনটির ভিতরে মিথ্যা হয়েছে, আর মিথ্যাকে উল্টিয়ে (! নট) দিয়ে এর মান সত্যি হয়েছে।

\$x=5

\$y=9

!(\$x<\$y) \Rightarrow মিথ্যা

!(\$x>100) \Rightarrow সত্য

সত্য \Rightarrow মিথ্যা

মিথ্যা \Rightarrow সত্য

!(2>5 && 6<10)



সত্য

সত্য

মিথ্যা

ভিতরে সত্য বাইরের (! নট)দ্বারা মিথ্যা হয়েছে।

```
<?php
$x = 300;
$y = 100;
var_dump !($x<$y || 5>20);
?>
```

প্রোথাম নং ১৩৩ (p133.php) Logical অপারেটর এর ব্যবহার

```
<?php
$x = 50;
$y = 10;
var_dump !($x>$y);
?>
```

প্রোথাম নং ১৩৪ (p134.php): Logical অপারেটর এর ব্যবহার

```
<?php
$x = 45;
$y = 95;
$p=5 ;
$q= 10 ;
var_dump !($y > $x ); ?>
```

প্রোগ্রাম নং ১৩৫ (p135.php): Logical অপারেটর এর ব্যবহার

```
<?php
$x = 50;
$y = 10;
var_dump !($y < $x || $x==50);
?>
```

লজিক্যাল অপারেটরস (Logical Operators)

Operator	Description	Example
&&	And	x=6 y=3 (x < 10 && y > 1) returns true
	Or	x=6 y=3 (x==5 y==5) returns false
!	Not	x=6 y=3 !(x==y) returns true

C-156

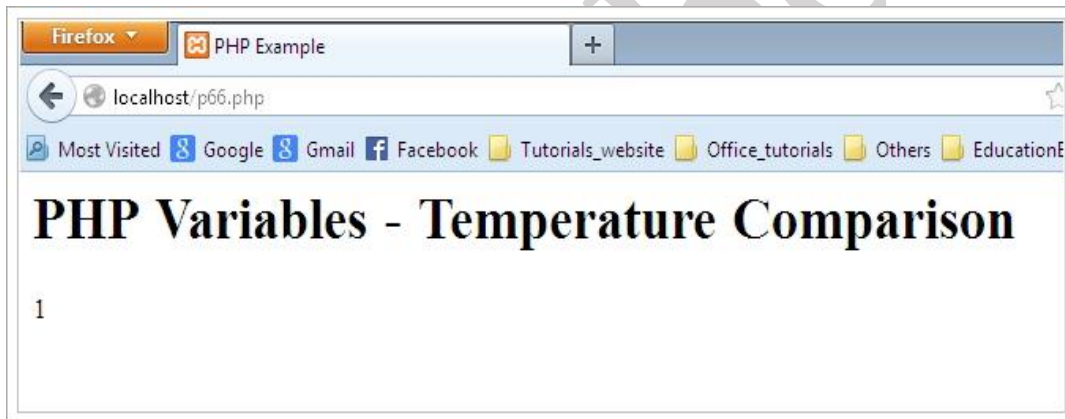
```
<html> <head> <title>PHP Example</title> </head>
<body>
<?php
$a = 8;
$b = 3;
$c = 3;
echo ($a == $b) && ($c > $b),"<br>";
echo ($a == $b) || ($b == $c),"<br>";
echo !($b <= $c),"<br>";
?>
</body> </html>
```

উপরের প্রোগ্রামের Output: 1

প্রোগ্রাম নং ১৩৬ (p136.php) Logical অপারেটর এর ব্যবহার

```
<html>
<head><title>PHP Example</title></head>
<body>
<h1>PHP Variables - Temperature Comparison</h1>
```

```
<?php
$insideTemp = 10;
$outsideTemp = 20;
$compareTemp = false;
$currentMonth = "february";
$experimentMonth = "june";
$compareTemp = (($insideTemp > $outsideTemp)
    && ($currentMonth == $experimentMonth));
echo $compareTemp;
$compareTemp = (($insideTemp < $outsideTemp)
    || ($currentMonth == $experimentMonth));
echo $compareTemp;
?>
</body></html>
```



চিত্রঃ ৬.৮ প্রোগ্রাম p136 এর আউটপুট

প্রোগ্রাম নং ১৩৭ (p137.php) Logical অপারেটর এর ব্যবহার

```
<?php
$a = 0;
$b = 'G';
$c = false;
// 3 lines added ...
var_dump((int) 'G'); // => 0
var_dump((bool) 0); // => false
var_dump((bool) 'G'); // => true
echo ($a == $b) ? "true\n" : "false\n"; // 0 == 0 => true
echo ($a == $c) ? "true\n" : "false\n"; // false == false => true
echo ($c == $b) ? "true\n" : "false\n"; // false == true => false
// 3 lines added ...
```

```
echo ($a === $b) ? "true\n" : "false\n"; // => false
echo ($a === $c) ? "true\n" : "false\n"; // => false
echo ($c === $b) ? "true\n" : "false\n"; // => false
?>
```



চিত্রঃ ৬.৮ প্রোগ্রাম p137 এর আউটপুট

প্রোগ্রাম নং ১৩৮ (p138.php): Logical অপারেটর এর ব্যবহার

```
<?php
$a = 3;
$b = 3;
$c = 10;
// all of the following are true
echo ($a == $b);
echo ($a != $c);
echo (8 === 8);
echo ("8" !== 8);
echo ($b < $c);
echo ($b <= $b);
echo ($c > $b);
echo ($c >= $c);
?>
```



চিত্রঃ ৬.৮ প্রোগ্রাম p138 এর আউটপুট

প্রোগ্রাম নং ১৩৯ (p139.php) Logical অপারেটর এর ব্যবহার

```
<?php
// -----
// foo() will never get called as those operators are short-circuit
$a = (false && foo());
$b = (true || foo());
$c = (false and foo());
$d = (true or foo());
// -----
// "||" has a greater precedence than "or"
// The result of the expression (false || true) is assigned to $e
// Acts like: ($e = (false || true))
$e = false || true;
// The constant false is assigned to $f and then true is ignored
// Acts like: (($f = false) or true)
$f = false or true;
var_dump($e, $f);
// -----
// "&&" has a greater precedence than "and"
// The result of the expression (true && false) is assigned to $g
// Acts like: ($g = (true && false))
$g = true && false;
// The constant true is assigned to $h and then false is ignored
// Acts like: (($h = true) and false)
$h = true and false;
var_dump($g, $h);
?>
```



চিত্রঃ ৬.৮ প্রোগ্রাম p139 এর আউটপুট

প্রোগ্রাম নং ১৪০ (p140.php) Logical অপারেটর এর ব্যবহার

```
<?php
// foo() will never get called as those operators are short-circuit
$a = (false && foo());
$b = (true || foo());
$c = (false and foo());
$d = (true or foo());
// "||" has a greater precedence than "or"
$e = false || true; // $e will be assigned to (false || true) which is true
$f = false or true; // $f will be assigned to false
var_dump($e, $f);
// "&&" has a greater precedence than "and"
$g = true && false; // $g will be assigned to (true && false) which is false
$h = true and false; // $h will be assigned to true
var_dump($g, $h);
?>
```



চিত্রঃ ৬.৮ প্রোগ্রাম p140 এর আউটপুট

প্রোগ্রাম নং ১৪১ (p141.php) Logical অপারেটর এর ব্যবহার

```
<?php
$a=TRUE; //1
$b=FALSE;//null 0
Echo "A has:". (bool)$a;
echo "B has:". (bool)$b;
echo "A AND B has: ";
echo $a && $b;
echo "A OR B has: ";
echo $a || $b;
?>
```

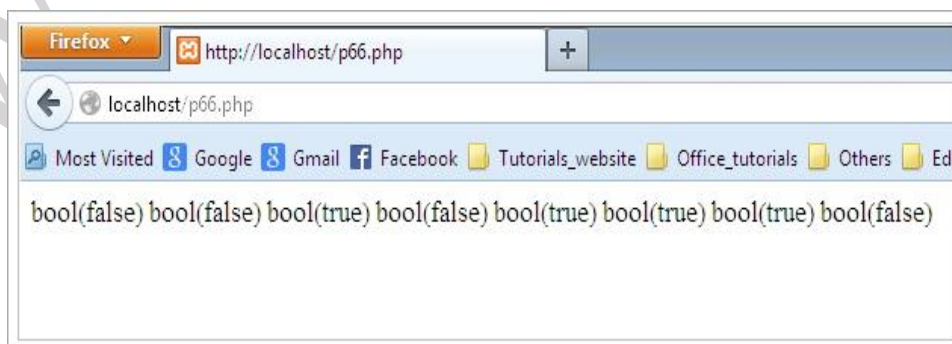


চিত্রঃ ৬.৮ প্রোগ্রাম p141 এর আউটপুট

প্রোগ্রাম নং ১৪২ (p142.php) Logical অপারেটর এর ব্যবহার

```
<?php
    $a = true && false;
    var_dump($a);
    $b = false && true;
    var_dump($b);
    $c = true && true;
    var_dump($c);
    $d = false && false;
    var_dump($d);
    $a = true || false;
    var_dump($a);
    $b = false || true;
    var_dump($b);
    $c = true || true;
    var_dump($c);
    $d = false || false;
    var_dump($d);
```

?>



চিত্রঃ ৬.৮ প্রোগ্রাম p142 এর আউটপুট

প্রোগ্রাম নং ১৪৩ (p143.php) Logical অপারেটর এর ব্যবহার

```
<?php
$a=12;
$b=0;
$c=22;
$d=22;
echo "\$a && \$b =";
var_dump ($a && $b);
echo "<br/>";
echo "\$c && \$d =";
var_dump ($c && $d);
echo "<br/>";
echo "\$a || \$b =";
var_dump ($a || $b);
echo "<br/>";
echo "\$c || \$d =";
var_dump ($c || $d);
echo "<br/>";
?>
```

উপরের ১৪৩নং প্রোগ্রামের Output:

```
$a && $b =bool(false)
$c && $d =bool(true)
$a || $b =bool(true)
$c || $d =bool(true)
```

প্রোগ্রাম নং ১৪৪ (p144.php) Logical অপারেটর এর ব্যবহার

```
<?php
$a=0;
$b=45;
var_dump ($a and $b);
echo "<br/>";
$a=10;
$b=45;
var_dump ($a and $b);
echo "<br/>";
$a=NULL;
$b=45;
var_dump ($a and $b);
```

```
echo "<br/>";
$a=10;
$b=45;
unset($b);
var_dump ($a and $b);
echo "<br/>";
$a=0;
$b=45;
var_dump ($a or $b);
echo "<br/>";
$a=10;
$b=45;
var_dump ($a or $b);
echo "<br/>";
$a=NULL;
$b=45;
var_dump ($a or $b);
echo "<br/>";
$a=10;
$b=45;
unset($b);
var_dump ($a or $b);
echo "<br/>";
?>
```

উপরের ১৪৪নং প্রোগ্রামের : Output:

```
bool(false) bool(true)
bool(false) bool(false)
bool(true) bool(true)
bool(true) bool(true)
```

৬.৯ অধ্যায় সারসংক্ষেপ :

> খুব সাধারণভাবে বলতে হয় প্রোগ্রামিং এর প্রত্যেকটি লাইন (Line of code) কে আমরা এক্সপ্রেশন বলতে পারি যেমন \$a= 6; \$b=2+3; echo "Dhaka"; echo \$b; ইত্যাদি এখানে আমি চারটি এক্সপ্রেশন লিখেছি। অপারেটর কাজ করে হচ্ছে অপারান্ড (Operand) নিয়ে \$k=2 এখানে \$k এবং 2 হচ্ছে অপারান্ড আর ইকুয়াল চিহ্নটি হচ্ছে অপারেটর।

> পিএইচপিতে বিভিন্ন ধরনের অপারেটর রয়েছে যেমন অ্যারিথমেটিক, রিলেশনাল, লজিক্যাল, অ্যাসাইনমেন্ট, ইনক্রিমেন্টাল, ডিক্রিমেন্টাল, কন্ডিশনাল, বিটওয়াইজ ইত্যাদি। অ্যারিথমেটিক অপারেটর গুলো যেমন + (Addition), - (Subtraction), * (Multiplication), / (Division), % (Modulus) ডিভিশন (/) অপারেটর এর কাজ হচ্ছে দুটো নাম্বার এর ভাগফল দেওয়া। যেমন 6/3 এখানে 6 কে 3 দিয়ে ভাগ দিবে এবং ভাগফল দুই হবে অর্থাৎ $a = 6/3$ এখানে a এর মান হবে 2।

> অ্যাসাইনমেন্ট অপারেটর ব্যবহার করে কোন ভ্যারিয়েবল এর ভ্যালুকে সেট করার জন্য। কম্প্যারিজন অপারেটর ব্যবহার করে দুটো অপারেন্ড এর মধ্য তুলনা করা হয় যেমন $5 == 3$ এখানে বুঝানো হয়েছে 5 কি 3 এর সমান? উত্তর হল না। কম্প্যারিজন অপারেটর গুলো হল ==, !=, >, <, >=, <=, ইত্যাদি। লজিক্যাল অপারেটর ব্যবহার করা হয় কোন এক্সপ্রেশনকে ট্রেস্ট করার জন্য। লজিক্যাল অপারেটর সমূহ &&, ||, !

৬.১০ প্রশ্ন

১. অপারেটর বলতে কি বুঝ?
২. প্রোগ্রামিং এ এক্সপ্রেশন বলতে কি বুঝ?
৩. বিভিন্ন ধরনের অপারেটর গুলোর নাম লিখ।
৪. মডিউলাস অপারেটর কি একটি উদাহরণ দাও।
৫. ইনক্রিমেন্ট এবং ডিক্রিমেন্ট অপারেটর এর মধ্য পার্থক্য লিখ।
৬. মাইনাস ইকুয়াল অপারেটর কিভাবে কাজ করে উদাহরণ দাও।
৭. অ্যারিথমেটিক অপারেটর গুলো উদাহরণ সহ লিখ।
৮. ইজ-ইকুয়ালিটি অপারেটরটির ফ্ল্যাগ-চাটঅ লিখ।
৯. ইজ লেজদেন অপারেটরটির ফ্ল্যাগ-চাট উদাহরণ সহ লিখ।
১০. কম্প্যারিজন অপারেটর গুলো উদাহরণ সহ লিখ।
১১. এন্ড এবং অর অপারেটর এর পার্থক্য লিখ।
১২. একটি ক্লাস লিখ op নামে এবং ঐ ক্লাসের মাঝে B নামে একটি ফাংশন আছে। B ফাংশন এর ভিতর তুলনা করে দেখাও ৫ বড় নাকি ৯ বড়।
১৩. একটি ক্লাস লিখ book এবং এই ক্লাসের ভিতর Pen নামে একটি ফাংশন আছে। Pen ফাংশন এ P নামক একটি ভ্যারিয়েবল আছে। P এর মান হচ্ছে ৪ এখন P কে ইনক্রিমেন্ট করে এক বাড়াও এবং P কে প্রিন্ট কর।

অধ্যায়-১২ মেথড (Methods)

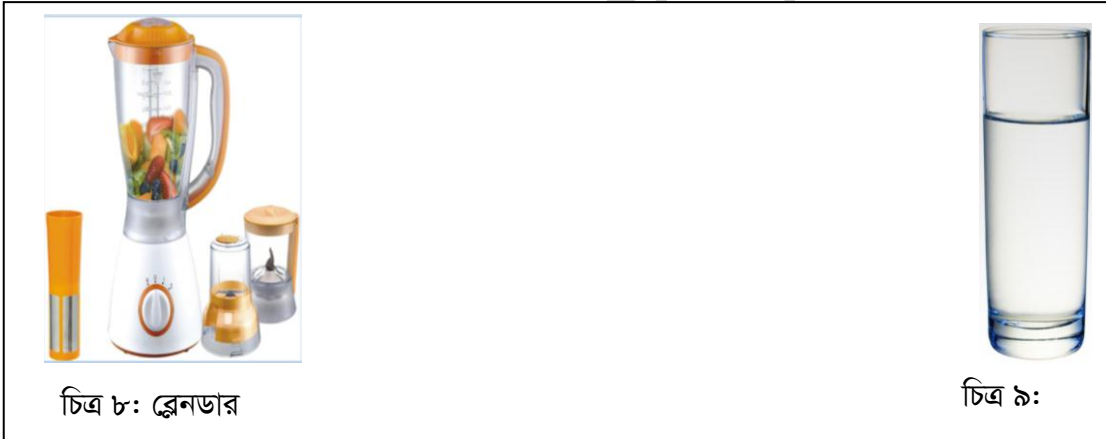
- ১২. মেথড (Methods) -----
- ১২.১ ভূমিকা -----
- ১২.২ মেথড কি (Method) -----
- ১২.৩ মেথড সিনটাক্স -----
- ১২.৪ মেথড তৈরী -----
- ১২.৫ মেথড কল -----
- ১২.৬ মেথড প্যারামিটার -----
- ১২.৭ মেথড প্যারামিটার পাঠানো -----
- ১২.৮ মেথড রিটার্ন টাইপ -----
- ১২.৯ অধ্যায় সারাংশ -----
- ১২.১০ অনুশীলন প্রোগ্রাম -----
- ১২.১১ প্রশ্নমালা -----

১২.১ ভূমিকা :

মেথড হচ্ছে এক ধরনের ফাংশনের মত তবে অবজেক্ট অরিয়েন্টেড প্রোগ্রামে মেথড ক্লাস এর সাথে সংযুক্ত (Associate) থাকে। মেথড সাধারণত অবজেক্ট অরিয়েন্টেড (অঅ) এর ক্ষেত্রে ব্যবহার করা হয়। মেথডকে ক্লাসের একটি অংশ (part) বলা যায়। একটি জামা আপনি পরিধান করেন আর আপনার হাত আপনার সাথে। অর্থাৎ এখানে হাত টা কে মেথড এর সাথে তুলনা করতে পারি আর জামাটাকে ফাংশন এর সাথে তুলনা করতে পারি।

১২.২ মেথড কি (What is Method)?

মেথড হচ্ছে এক ধরনের ফাংশন। নির্দিষ্ট কিছু কাজ মেথড করে থাকে। অর্থাৎ মেথড সব সময় একই ধরনের কাজ করে থাকে যার জন্য একই ধরনের কোন কাজ করার জন্য যেন একই কোড বার বার লিখতে না হয়। সে জন্য আমরা মেথড ব্যবহার করতে পারি। মনে রাখবেন আমরা চেষ্টা করছি লেখাগুলোকে প্রচলিত শব্দে লিখতে সে জন্য হয়তো সঠিক শব্দ ব্যবহার হচ্ছে না সে জন্য আমরা দুঃখিত। মেথডকে সাধারণত আমরা ক্লাসের ভিতরে লিখে থাকি। আপনি চাইলে কোন তথ্য ক্লাস এর বাইরে থেকে ক্লাসের মেথড এর ভিতর পাঠাতে পারেন। তবে মনে রাখতে হবে মেথড অথবা ফাংশন নিজে নিজে রান (execute) করে না যতক্ষণ না পর্যন্ত এটিকে কল (call) করা না হয়। মেথড অবজেক্ট কোন একটি নির্দিষ্ট কাজ করতে সাহায্য করে। উদাহরণ হিসেবে বলা যায় মাউস ক্লিক অনেকটা মেথড এর মত। যেমন আমরা যখন ফোল্ডার ওপেন করি মাউস ক্লিক করি, যখন কোন ফাইল ওপেন করি মাউস ক্লিক করি, যখন কোন ব্রাউজার ওপেন করি মাউস দিয়ে ক্লিক করি অর্থাৎ এখানে সব কাজের জন্য মাউস ক্লিক একটি কমন কাজ।



একই মেশিন ব্যবহার করে আপনি অনেক ধরনের জুস তৈরি করতে পারেন ঠিক তেমনি একই গ্লাস ব্যবহার করে অনেকে কিছু পান করতে পারেন। এমনি অনেকে একই গ্লাস ব্যবহার করতে পারেন।

অবশ্যই মনে রাখবেন এই বইটি পড়ার পূর্বে আমাদের লেখা পিএইচপি অ্যান্ড মাইএসকিউল বইটি অবশ্যই পড়বেন। অন্যথায় এই বইটি বুঝতে আপনার সমস্যা হতে পারে।

১২.৩ মেথড সিনট্যাক্স (Syntax)

যেহেতু মেথড ক্লাসের ভিতর থাকে সেজন্য প্রথমে যে কোন একটি ক্লাস নিতে হবে। ক্লাসের নাম যে কোন কিছু হতে পারে। তার পর Function কী-ওয়ার্ডটি ব্যবহার করতে হবে। চাইলে Function এর পূর্বে Public কী-ওয়ার্ডটি ব্যবহার করতে পারেন। তারপর Function এরপর মেথড এর যে কোন একটি নাম দিতে হবে। এরপর দু'টো কার্লি (Second) ব্রাকেট দিয়ে মেথড এর বডি শেষ করতে হবে।


```
class food{  
    public Function mango ()  
    {  
        Method body  
    }  
}
```

১২.৩.১ মেথড সিনট্যাক্স

নিম্নে মেথড সিনট্যাক্স এর ধাপ সমূহ

১. Class ক্লাস নেইম
২. সেকেন্ড ({}) ব্রাকেট
৩. পাবলিক (Public) কী-ওয়ার্ড
৪. ফাংশন (Function) কী-ওয়ার্ড
৫. মেথড এর নাম (যে কোন নাম)
৬. দু'টো প্রথম ব্রাকেট ()
৭. সেকেন্ড ব্রাকেট শুরু
৮. মেথড এর বডি (Block of code)
৯. সেকেন্ড ব্রাকেট শেষ (৭ এর জন্য)
১০. সেকেন্ড ব্রাকেট শেষ (২ এর জন্য)

১২.৪ মেথড তৈরী (Create Method)

প্রোগ্রামে ফাংশন লিখলেই তা কাজ করবে না। ফাংশনকে execute (কাজ) করাতে হলে, এটিকে কল করতে হবে। আর কল করার নিয়ম হচ্ছে শুধু ফাংশন এর নাম লিখে প্রথম ব্রাকেট ব্যবহার করতে হবে। যেমন আপনার ফাংশনের নাম display. তাহলে কল করার নিয়ম হল display()

ফাংশন নেইম dhaka

কল করতে হবে dhaka()

ফাংশন নেইম beauty

কল করতে হবে beauty()

প্রোগ্রাম নং ৩৭৯ (p379.php) Function এর ব্যবহার

```
<?php  
function Fruit() {  
    echo "My favorite fruit is pomegranate."  
}  
Fruit();  
?>
```

উপরের ৩৭৯নং প্রোগ্রামের Output: My favorite fruit is pomegranate.

উপরের প্রোগ্রামে ফাংশনটির নাম হল Fruit তাই Fruit() লিখে কল করা হয়েছে। আর আউটপুটে শো করবে My Favourite fruit is pomegranate

১২.৫ মেথড কল

মেথড যেহেতু ক্লাসের ভিতর থাকে সেজন্য মেথডকে ব্যবহার (Access) করতে হলে ঐ ক্লাসের অবজেক্ট তৈরী করতে হবে। তারপর ক্লাসের অবজেক্ট দিয়ে মেথডকে কল করতে হবে। তখন মেথড এর ভিতরের কোড গুলো কাজ (রান) করবে। নিম্নে Student নামক ক্লাসের মেথডকে কল করে দেখানো হয়েছে।

```
<?php
Class Student {
    public Function subject() {
        echo "project management";
    }
}
std= new student;
std-> subject (); মেথডকে কল করে দেখানো হয়েছে।
?>
```

উপরের প্রোগ্রামটি রান করলে আউটপুট হবে project management

```
<?php
Class teacher
    $age=27; {
        public Function te (){
            echo "age"; }
    }
    $t = new teacher ();
    $t -> te (); মেথডকে কল করে দেখানো হয়েছে।
?>
```

উপরের প্রোগ্রামের : আউটপুট : ২৭

১২.৬ মেথড প্যারামিটার

আপনি বিভিন্ন সময়ে বিভিন্ন নাম্বার যোগ করতে চান সেজন্য বার বার কোড না লিখে, এমন ভাবে একবার কোড লিখে রাখেন যেন বিভিন্ন সময় নাম্বার (প্যারামিটার) পাঠালে মেথডটি যোগ করে দেয়। প্যারামিটার সাধারণত মেথড এ পাঠানো হয় ক্লাস এর বাইর থেকে। প্যারামিটার সাধারণত মেথডকে কল করার সময় পাঠানো হয়।

```
<?php
function pen ($name)
{
    echo "age";
}
```

অবজেক্ট অরিয়েটেড পিএইচপি

```
$xp -> pen ("Echono");
```

```
?>
```

এখানে “Echono” এটাকে প্যারামিটার হিসেবে pen মেথডে পাঠিয়েছি যেটি সেখানে \$name এ গিয়েছে তাই \$name ভ্যারিয়েবলটির মান হল Echo no ।

> একের অধিক প্যারামিটার

আপনি মেথডে একের অধিক বিভিন্ন ধরনের ভ্যালু প্যারামিটার হিসেবে পাঠাতে পারেন ।

```
<?php
```

```
function pen ($n, $p)
```

```
{
```

```
    echo $n;
```

```
    echo $p;
```

```
}
```

```
b -> pen ("Echo", 15);
```

```
?>
```

অর্থাৎ আপনি কমা দিয়ে (কমার পর স্পেস) একাধিক ভ্যালু মেথডে পাঠাতে পারেন । এখানে \$n এর মান হবে Echo আর \$p এর মান হবে ১৫ ।

```
<?php
```

```
class p{
```

```
    function pen($n, $r){ // inside bracket just write variable as much as want
```

```
        echo $n;
```

```
        echo $r;
```

```
    }
```

```
}
```

```
$g = new p;
```

```
$g->pen("echo no", 5); // first values goes to $n first variable
```

```
?>
```

```
<?php
```

```
class p{
```

```
    function pen($name, $customer) {
```

```
        echo $name;
```

```
        echo $customer;
```

```
    }
```

```
}
```

```
$g = new p;
```

```
$g->pen("echo no", "kamal");
```

```
?>
```

অবজেক্ট অরিয়েটেড পিএইচপি

আপনি যদি মেথডে কোন ভ্যালু পাঠাতে চান তাহলে কি ভাবে পাঠাবেন। সেজন্য মেথড এর ব্রাকেটের () এর ভিতর ভেরিয়েবল লিখতে হবে। আর মেথড কল করার সময় ভ্যালুটি দিয়ে দিতে হবে।

penType(\$price) ← এটি হচ্ছে প্যারামিটার
\$obj->penType(2) ← প্যারামিটার ভ্যালু

প্রোগ্রাম ৩৮০ p380.php: মেথড প্যারামিটার

```
<?php
class pen {
    function PenType($price){ // $price is parameter its vlaue come when
call it
    echo $price; // display the value of $price
    }
}
$obj =new pen();
$obj->PenType(2); // passing value of $price, value is 2
?>
```

উপরের ৩৮০ নং প্রোগ্রামের : আউটপুট: 2

প্রোগ্রাম ৩৮১ (p381.php): একের অধিক প্যারামিটার

```
<?php
class pen {
    function PenType($price, $penName){ // if more than one parameter
write using comma
    echo $price; // display the value of $price
    echo $penName;
    }
}
$obj =new pen();
$obj->PenType(13, "Econo"); // if want to pass more paramter value
write using comma
?>
```

উপরের ৩৮১ নং প্রোগ্রামের : আউটপুট: 13Econo

প্রোগ্রাম ৩৮২ (p382.php):Parameter পাঠানো

```
<?php
class pen {
    function PenType($price, $penName, $companyName){ // if more than
one parameter write using comma
        echo $price; // display the value of $price
        echo $penName;
        echo $companyName;
    }
}
$obj =new pen();
$obj->PenType(13, "Econo", "Ocan"); // if want to pass more paramter
value write using comma
?>
```

উপরের ৩৮২ নং প্রোগ্রামের ৪ আউটপুট: 13EconoOcan

১২.৮ মেথড রিটার্ন (Return)

মেথড অনেক সময় ডেটা প্রসেস করে অথবা এমনিতে কোন কিছু রিটার্ন করতে পারে। অর্থাৎ মেথড ক্লাসের বাইরে কোন ভ্যালু পাঠায় যেখান থেকে মেথডকে কল করা হয়েছে। আর এই প্রক্রিয়াটিই হচ্ছে মেথড রিটার্ন ভ্যালু। রিটার্ন করার জন্য রিটার্ন কী-ওয়ার্ডটি ব্যবহার করতে হবে।

```
<?php
function mm() {
    return 10; // এখানে return হচ্ছে কী-ওয়ার্ড
}
?>
```

অর্থাৎ return কী-ওয়ার্ড এ যা লিখবেন সেটি মেথড পাঠাবে।

```
<?php
function mn($p) {
    return $p;
}
?>
```

এখানে return করছে \$p এর মানকে।

```
<?php
function mp()
{
    return "Bookbd series";
}
```

```
}  
?>
```

এখানে একটি স্ট্রিং রিটার্ন করছে।

```
<?php  
function Sum()  
{  
    $p= 5+3;  
    return "$p";  
}  
?>
```

মেথড থেকে আমরা কিভাবে কোন রেজাল্ট পেতে পারি। অর্থাৎ মেথড এর কোন ভ্যালু আমরা নিতে চাই যেটা বলে মেথড রিটার্ন। অর্থাৎ মেথড কোন একটি কাজ করে তার ভ্যালুটা যদি ক্লাস অথবা মেথড এর বাইরে পাঠায় সেটাই হল মেথড রিটার্ন ভ্যালু।

মেথড ভ্যালু পাঠানোর (Return) জন্য Return কী-ওয়ার্ডটি ব্যবহার করে। অর্থাৎ Return 5; এখানে 5 কে পাঠাচ্ছে যেখান থেকে কল করা হচ্ছে সেখানে।

প্রোগ্রাম ৩৮৩ (p383.php): রিটার্ন মেথড ভ্যালু

```
<?php  
class pen {  
    function PenType() {  
        return 5; // return value 5 // use return keyword to reutrn vlaue  
    }  
}  
$obj =new pen();  
echo $obj->PenType(); // Here we will get 5 becuase call  
PenType from here  
?>
```

উপরের ৩৮৩ নং প্রোগ্রামের ৪ আউটপুট: 5

প্রোগ্রাম ৩৮৪ (p384.php):মেথড রিটার্ন ভ্যালু

```
<?php  
class pen {  
    function PenType() {  
        return 45.23; // return value 5 // use return keyword to reutrn  
vlaue
```

```
    }  
  }  
  $obj =new pen();  
  echo $obj->PenType(); // Here we will get 5 because call  
  PenType from here  
?>
```

উপরের ৩৮৪ নং প্রোগ্রামের : আউটপুট: 45.23

প্রোগ্রাম ৩৮৪ (p384.php): স্টিং রিটার্ন

```
<?php  
class pen {  
    function PenType() {  
        return "Econo"; // return string type value like Econo  
    }  
}  
$obj =new pen();  
echo $obj->PenType(); // Here we will get Econo  
?>
```

উপরের ৩৮৪ নং প্রোগ্রামের : আউটপুট: Econo

প্রোগ্রাম ৩৮৫ (p385.php) :মেথড রিটার্ন ভেরিয়েবল

```
<?php  
class pen {  
    public $price=50;  
    function PenType() {  
        return $this->price; // return string type value like Econo  
    }  
}  
$obj =new pen();  
echo $obj->PenType(); // Here we will get Econo  
?>
```

উপরের ৩৮৫ নং প্রোগ্রামের : আউটপুট: 50

```
<?php  
class marks {  
    function PenType() {
```

অবজেক্ট অরিয়েটেড পিএইচপি

```
        return 6;
    } }
$b= new marks;
echo $b-> test(); // আউটপুট হবে ৬
$k = $b -> test(); // এখানে K এর মান হল ৬
echo $k; // আউটপুট হবে ৬
?>
```

একটি ক্লাসে একের অধিক মেথড থাকতে পারে তবে ভিন্ন নামে। যেমন একটি ক্লাসে তিনটি মেথড আছে orange, green এবং blue.

```
<?php
class marks {
    function Orange() {
    }
    function Green() {
    }
    function Blue() {
    }
}
?>
```

এখন আপনি Color ক্লাসের একই অবজেক্ট দিয়ে সব গুলো মেথডকে আপনার প্রয়োজনে কল করতে পারবেন। যেমনঃ \$dark = new color; এখানে \$dark হচ্ছে Color ক্লাসের একটি অবজেক্ট। এখন \$dark অবজেক্ট দিয়ে Blue মেথডকে কল করবো।

```
$dark -> blue();
```

এবার \$dark অবজেক্ট দিয়ে Green মেথডকে কল করবো (আপনি চাইলে নতুন আরেকটি অবজেক্ট তৈরী করতে পারেন)।

```
$dark -> Green ();
```

এবার Color ক্লাসের আরেকটি অবজেক্ট তৈরী করবো (একটি ক্লাসের একাধিক অবজেক্ট তৈরী করা যায়)

```
$c = new color;
```

মনে রাখবেন অবজেক্ট এর নাম আপনি যে কোন কিছু লিখতে পারেন।

```
<?php
class p {
    function pen($n, $r) {
        echo $n;
        echo $r;
    }
    function pen($k) {
```


অবজেক্ট অরিয়েটেড পিএইচপি

```
        echo $k;
    }
}
$g = new p;
$g->pen("echo no", 5);
$g->pn(10);
?>
```

উপরের প্রোগ্রামের : আউটপুট : echo no 5 এবং ১০

```
<?php
class color{
    function green() {
        echo "This is Green Color";
    }
    function red() {
        echo "This is Red Color";
    }
    function blue() {
        echo "This is Blue Color";
    }
}
$dark = new color;
$dark->red(); // here we call red method using dark object
?>
```

উপরের প্রোগ্রামের : আউটপুট : This is red color

```
<?php
class color {
    function green() {
        echo "This is Green Color";
    }
    function red() {
        echo "This is Red Color";
    }
    function blue() {
        echo "This is Blue Color";
    }
}
```

```
}  
$dark = new color;  
$dark->red(); // here we call red method using dark object  
$dark->green(); // call green method using dark it can be new or different  
object  
?>
```

উপরের প্রোগ্রামের : আউটপুটঃ This is Red Color, This is Green Color

১২.৯ অধ্যায় সারাংশ

> মেথড হচ্ছে এক ধরনের বিশেষ ফাংশন। ফাংশন ক্লাস ছাড়া লিখা যায় কিন্তু মেথডকে ক্লাসের ভিতর লিখতে হয়। অর্থাৎ ক্লাসের কোন নির্দিষ্ট ফাংশনকে মেথড বলা যায়। ক্লাস এর বাইর থেকে আমরা কোন ভ্যালু (প্যারামিটার) মেথড এর সাহায্যে ক্লাস এর ভিতরে পাঠাতে পারি। ফাংশন যেভাবে লিখতে হয় মেথড ও একই ভাবে লিখা যায়।

> মেথড যে ক্লাসে থাকে ঐ ক্লাসের অবজেক্ট দিয়ে মেথডকে কল করতে হয় মেথড নিজে নিজে রান করবেনা যতক্ষণ মেথডকে কল করা না হয়। মেথড এ চাইলে ভ্যালু (ডেটা) প্যারামিটার হিসেবে পাঠাতে পারেন। মেথড ভ্যালু গুলো গ্রহন করে। একটি ক্লাসে একের অধিক মেথড থাকতে পারে। যখন যেই মেথড কল করবেন তখন সেটি কাজ করবে। মেথড আবার ভ্যালু রিটার্ন করে। প্রশ্ন কোথায় মেথড রিটার্ন ভ্যালু পাঠায় উত্তর হল যেখানে থেকে মেথডকে কল করা হয়েছিল সেখানে।

১২.১০ প্রশ্ন

১. পিএইচপি প্রোগ্রামিং এ মেথড বলতে কি বুঝ?
২. মেথড এবং ফাংশনের মধ্য পার্থক্য লিখ।
৩. মেথড এর সিনট্যাক্সটি লিখ।
৪. একটি ক্লাসে একটি মেথড লিখ। মেথডটিতে Bangladesh echo কর। তারপর মেথডটিকে কল করে দেখাও।
৫. মেথড প্যারামিটার বলতে কি বুঝ?
৬. একটি ক্লাস হচ্ছে pen এ দু'টি মেথড red এবং blue। red মেথড ডিসপ্লে (echo) করে red pen কিন্তু blue মেথড রিটার্ন করে ৭৫। প্রোগ্রামটি লিখ।

QUICK & EASY WAY

Your Professional Trainer...
নিজে নিজে এসইও শিখুন

অ্যাডভান্স
সার্চ ইঞ্জিন অপটিমাইজেশন
From Advance to Professional

রচনা ও সম্পাদনা
বুকবর্ড ইনফো

কী-ওয়ার্ড অ্যানালাইসিস * সার্চ ইঞ্জিন সিক্রেট * এসইও টুলস * মেইল চিহ্ন
* গুগলের নতুন এসইও স্ট্র্যাটেজি * লিংক তৈরি * এসইও ট্রাভেল শূটিং

কিভাবে সবচেয়ে বড় সার্চ ইঞ্জিনকে কন্ট্রোল করে বিশ্বের মাঝে আপনার ওয়েব
সাইটকে বেশি পরিচিতি করা যায় এ পুরো বিষয়টি বইটি পড়ে খুব সহজে বুঝবেন।
অল্প শিক্ষিত এবং টেকনিক্যাল পার্সন খুব সহজে এমইও শিখতে পারবেন।

ডিজিটাল ডিভিডিও কন্টেন্ট

All Updated Tools & Techniques of SEO

ডাউনলোড

bookbd.info

অধ্যায় ১৩. মডিফায়ার এবং কী-ওয়ার্ড (Modifiers)

১৩. মডিফায়ার এবং কী-ওয়ার্ড (Modifiers)

১৩.১ ভূমিকা -----

১৩.২ মডিফায়ার কি? -----

১৩.৩ মডিফায়ার লিখার নিয়ম -----

১৩.৪ পাবলিক (public) -----

১৩.৫ প্রাইভেট (Private) -----

১৩.৬ প্রোটেক্টেড (Protected) -----

১৩.৭ অ্যাবসট্রাক্ট এবং ফাইনাল (Abstract Final) -----

১৩.৮ এক নজরে অধ্যায় -----

১৩.৯ প্রশ্ন -----

১৩.১ ভূমিকাঃ

মনে করুন আপনার একটি মাইক্রোবাস আছে এখন প্রশ্ন হচ্ছে এটি কি শুধু ঢাকাতে চলবে। নাকি ঢাকার বাহিরে ও চালানো যাবো এ রকম একটি বিষয় হচ্ছে প্রোগ্রামিং এ মডিফায়ার। প্রোগ্রামিং এ ভ্যারিয়েবল, ফাংশন, মেথড, ক্লাস ইত্যাদিও ব্যবহার (Access) এর সীমাবদ্ধতা করে দেওয়াই হচ্ছে মডিফায়ার এর কাজ। যেমন একটি ভ্যারিয়েবল নিজের ক্লাসের বাহিরে অন্য ক্লাস ব্যবহার করতে পারবে কিনা। কিছু মডিফায়ার (Modifier) যেমন পাবলিক (Public), প্রাইভেট (Private) এবং প্রোটেক্টেড ইত্যাদি।

১৩.২ মডিফায়ার (Modifier) কি?

বিশেষ ধরণের কিছু কী ওয়ার্ড আমরা ব্যবহার করে থাকি ক্লাস, মেথড, ফাংশন এবং ভ্যারিয়েবল এর পূর্বে আর এটিকেই বলে modifier, অ্যাকসেসকে কন্ট্রল করার জন্য আমরা মডিফায়ার ব্যবহার করে থাকি। মডিফায়ার কে আবার visibility ও বলা হয়। modifier গুলো হচ্ছে যেমন public, private, protected, Final এবং Abstract

১৩.৩ মডিফায়ার লিখার নিয়ম

পূর্বেই বলেছি মডিফায়ার ভ্যারিয়েবল, মেথড, ক্লাস ইত্যাদির পূর্বে বসে।

Public \$a

a ভ্যারিয়েবলটি হচ্ছে পাবলিক

Private \$b

b ভ্যারিয়েবলটি হচ্ছে প্রাইভেট

Protected \$c

এখানে C ভ্যারিয়েবলটি হচ্ছে প্রোটেক্টেড

```
Public Function pen ()
```

```
{
```

```
}
```

এখানে pen মেথডটি হচ্ছে পাবলিক

```
Private Function color()
```

```
{
```

```
}
```

এখানে Color মেথডটি হচ্ছে প্রাইভেট

```
Private Class food
```

```
{
```

```
}
```

Food ক্লাসটি হচ্ছে প্রাইভেট মনে রাখবেন কোন কিছু না লিখলে সেটা বাই ডিফল্ট পাবলিক।

১৩.৪ পাবলিক (Public)

কোন ক্লাস, ভেরিয়েবল এবং ফাংশন এর পূর্বে পাবলিক কী-ওয়ার্ডটি থাকলে এটিকে আপনার স্ক্রিপ্ট এর যে কোন জায়গায় ব্যবহার করা যাবে। মনে রাখবেন ঐ ক্লাস অথবা প্রোপার্টিগুলো আপনার পিএইচপি ফাইলে সব জায়গায় ব্যবহার করা যাবে। যদি কোন ক্লাসের ভিতর মেথড অথবা ভেরিয়েবল পাবলিক, সেগুলোকে ক্লাসের ভিতরে এমনকি বাইরেও ব্যবহার করা যাবে।

```
Class pen
{
    public $b=3;
}
$0=new pen;
echo $0 -> b;
```

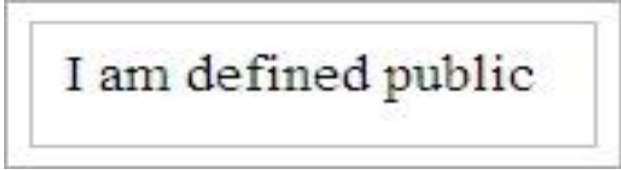
এখানে `b` এর মান ৩ দেখাবো এবং প্রোগ্রামিং এ কোন ভুল ও দেখাবেনা। `$b` এর পাবলিক না লিখলে ও হত কারণ কোন কিছু না থাকা মানে পাবলিক।

```
Class pen
{
    $b=3;
}
$0=new pen;
echo $0 -> b;
```

এখানে ও আউটপুট ও প্রদর্শন করবে। যদিও `$b` এর পূর্বে পাবলিক লিখা হয় নাই।

প্রোগ্রাম ৩৮৬ (p386.php): পাবলিক ভেরিয়েবল

```
<?php
class test {
    public $abc='I am defined public'; // abc variable defined by public
}
$objA = new test();
echo $objA->abc;//accessible from outside because abc defined by public
?>
```



I am defined public

চিত্রঃ ১৩.৪ প্রোগ্রাম p386 এর আউটপুট

প্রোগ্রাম ৩৮৭ (p387.php): পাবলিক মেথড

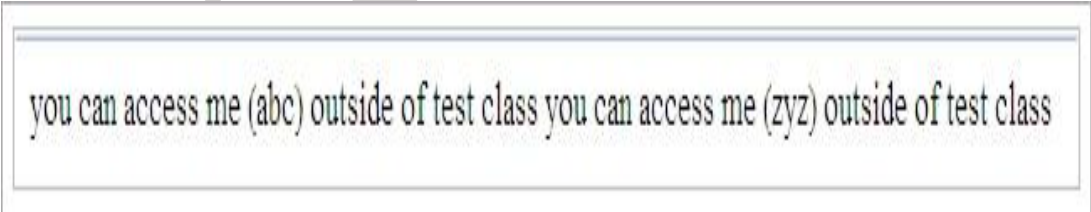
```
<?php
```

```
class test{
    public function xyz() // xyz method defined by public, xyz accessible
    from outside class {
        echo 'I am defined public method';
    }
}
$objA = new test();
$objA->xyz();//public method of the class test
?>
```

উপরের ৩৮৭ নং প্রোগ্রামের আউটপুট: I am defined public method

প্রোগ্রাম ৩৮৮ (p388.php): পাবলিক ভেরিয়েবল এবং মেথড

```
<?php
class test{
    public $abc='you can access me (abc) outside of test class ';
    public $xyz;
    public function xyz() {
        echo 'you can access me (zyz) outside of test class ';
    }
}
$objA = new test();
echo $objA->abc;//accessible from outside
$objA->xyz();//public method of the class test
?>
```



you can access me (abc) outside of test class you can access me (zyz) outside of test class

চিত্রঃ ১৩.৪ প্রোগ্রাম p388 এর আউটপুট

১৩.৫ প্রাইভেট (Private)

প্রাইভেট মেথড এবং প্রোপার্টি সমূহ শুধু মাত্র ক্লাসের ভিতরে ব্যবহার করার জন্য। প্রাইভেট ভ্যারিয়েবল এবং ফাংশনকে ক্লাসের বাইরে ব্যবহার করা যাবে না। তবে ক্লাসের মাঝে ব্যবহার করা যাবে \$this কী-ওয়ার্ড ব্যবহার করে। প্রশ্ন হল কেন ফাংশন অথবা ভ্যারিয়েবলকে প্রাইভেট করবো অর্থাৎ আপনি যদি কখনো মনে করেন আপনার ক্লাসের প্রোপার্টি ক্লাসের বাইরের কেউ ব্যবহার না করুক তখন প্রাইভেট ব্যবহার করতে হবে।

```
class book
```

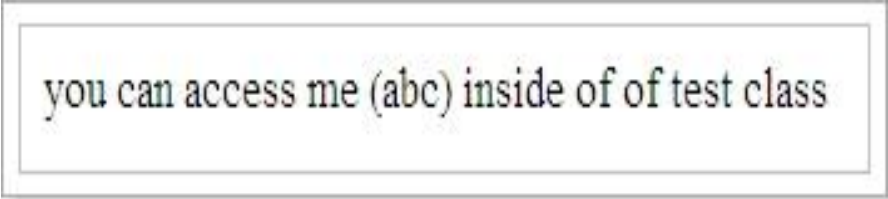


```
{  
    Private $p=15;  
}  
$b=new book;  
echo $b ->p;
```

এই প্রোগ্রামটিতে এরর (Error) দেখাবে। এররটি অনেকটা এরকম যে প্রাইভেট প্রোপার্টি অ্যাকসেস করা যাচ্ছে না (Can not Access private property) কিন্তু যদি এই \$p পাবলিক হত তাহলে কোন সমস্যা হত না।

প্রোগ্রাম ৩৮৯ (p389.php) প্রাইভেট ভ্যারিয়েবল

```
<?php  
class test{  
    private $abc='you can access me (abc) inside of of test class '  
    public function xyz(){  
        echo $this->abc;  
    }  
}  
$objA = new test();  
//echo $objA->abc;//accessible not possible from outside the test class  
$objA->xyz();//public method of the class test  
?>
```



you can access me (abc) inside of of test class

চিত্রঃ ১৩.৪ প্রোগ্রাম p389 এর আউটপুট

প্রোগ্রাম ৩৯০ (p390.php) প্রাইভেট ভ্যারিয়েবল এবং Constructor

```
<?php  
class Car {  
    private $NumberOfDoors;  
    public function __construct()  
    {  
        $this->NumberOfDoors = 4;  
        echo $this->NumberOfDoors;  
    }  
}
```

```
    }  
    // instantiate  
    $car= new Car();  
    // show the property - not allowed  
?>
```

উপরের ৩৯০নং প্রোগ্রামেরঃ আউটপুট: 4

১৩.৬ প্রোটেক্টেড (Protected)

প্রোটেক্টেড হচ্ছে অনেকটা পাবলিক এবং প্রাইভেট এর মাঝামাঝি। তবে প্রোটেক্টেড হচ্ছে প্রাইভেট এর মত তবে সেটিকে পিএইচপি ক্লাসে ব্যবহার করা যাবে ইনহেরিটেন্স এবং ইন্টারফেস এর ক্ষেত্রে। প্রিয় পাঠক, আমরা উপস্থাপন এবং বুঝানোর সুবিধার্থে অনেক সময় ইংরেজি বাংলা সংমিশ্রণ করে লিখছি যা ক্ষমা সুন্দর দৃষ্টিতে দেখবেন।

যেমন কোন ক্লাসের ভ্যারিয়েবল প্রোটেক্টেড হলে সেটা শুধুমাত্র ঐ ক্লাস অথবা তার চাইল্ড (Derived Class) ক্লাস ব্যবহার করতে পারবে। তবে কোন ভ্যারিয়েবল প্রাইভেট হলে সেটা ঐ ক্লাসে ব্যবহার যোগ্য কিন্তু চাইল্ড ক্লাসে ব্যবহার যোগ্য নয়।

```
Class ny  
{  
    Protected $b =6;  
}  
$z = new ny;  
$z->b; // Can not Access Protected Property.
```

উপরের প্রোগ্রামটিতে এরর দেখাবে কারণ প্রোটেক্টেড ভ্যারিয়েবলকে ক্লাস এর বাহিরে ব্যবহার (Access) করা যাবে না।

```
Class PC  
{  
    Protected $c =45;  
    // এখানে c হচ্ছে একটি ভ্যারিয়েবল তবে প্রোটেক্টেড করা আছে।  
}
```

```
Class mouse extends pc
```

```
{  
    এখানে Mouse ক্লাস হচ্ছে PC ক্লাসের সাব ক্লাস তাই Mouse ক্লাস চাইলে PC ক্লাসের পাবলিক এবং প্রোটেক্টেড ভ্যারিয়েবল এবং মেথড ব্যবহার করতে পারবে।
```

```
Function bd()
```

```
{  
    এটি হচ্ছে Mouse ক্লাস এর একটি মেথড
```

```
Echo $this -> c;
```

এখানে PC ক্লাসের c কে প্রিন্ট করছি তবে \$this কী-ওয়ার্ড ব্যবহার করতে হবে।

```
}
```

```
}
```

এখন যদি আমরা Mouse ক্লাস এর অবজেক্ট তৈরী করে bd কে কল করি তাহলে আউটপুট পাবো ৪৫।

```
$se = new mouse;
```

```
echo $se -> bd();
```

প্রোগ্রাম ৩৯১ (p391.php) প্রোটেক্টেড ভ্যারিয়েবল এক্সেস

```
<?php
```

```
class pc { // pc is parent class
```

```
    protected $pr=451; // define protected to stop access outside the class
```

```
    // but the child class can access the pr variable
```

```
}
```

```
class cc extends pc { // cc is child class is so it can access the pr
```

```
    public function testChild() {
```

```
        echo $this->pr; //will work because it cc is child of pc
```

```
    }
```

```
}
```

```
$objChild = new cc();
```

```
$objChild->testChild();//work because test child will call
```

```
?>
```

উপরের ৩৯১ নং প্রোগ্রামের আউটপুট: 451

প্রোগ্রাম ৩৯২ (p392.php) মডিফায়ার এর ব্যবহার

```
<?php
```

```
class parent{
```

```
    protected $pr;
```

```
    public $a
```

```
    protected function testParent(){
```

```
        echo this is test;
```

```
    }
```

```
}
```

```
class child extends parent{
```

```
    public function testChild(){
```

```
        $this->testParent(); //will work because it
```

```
    }
```

```
}
```

```
$objParent = new parent();  
$objParent->testParent();//Throw error  
$objChild = new Child();  
$objChild->setChild();//work because test child will call test parent.
```

?>

প্রোগ্রাম ৩৯৩ (p393.php) মডিফায়ার এর ব্যবহার

```
<?php  
class Myclass{  
    public $title;  
    public function DispTitle(){  
        echo $this -> title;  
        echo "<br />";  
    }  
}  
$obj = new Myclass();  
$obj->title = "ABCXYZ";  
echo $obj->title;  
echo "<br />";  
echo $obj->DispTitle();  
?  
?>
```

উপরের ৩৯৩ নং প্রোগ্রামেরঃ আউটপুট: ABCXYZ ABCXYZ

প্রোগ্রাম ৩৯৪ (p394.php) মডিফায়ার এর ব্যবহার

```
<?php  
class Myclass{  
    protected $title=10;  
}  
class Baseclass extends Myclass {  
    function __construct(){  
        echo $this->title;  
    }  
}  
$obj = new Baseclass();  
?  
?>
```

উপরের ৩৯৪ নং প্রোগ্রামেরঃ আউটপুট: 10

প্রোগ্রাম ৩৯৫ (p95.php) বিভিন্ন মডিফায়ার সমূহ

```
<?php
```

```
class myClass{
    public $public = 'Public';
    protected $protected = 'Protected';
    private $private = 'Private';
    function printHello() {
        echo $this->public;
        echo $this->protected;
        echo $this->private;
    }
}

$obj = new myClass();
echo $obj->public; // Works
echo $obj->protected; // Fatal Error
echo $obj->private; // Fatal Error
// Shows Public, Protected and Private
$obj->printHello();

class myClass2 extends myClass {
    // We can redeclare the public and
    // protected method, but not private
    protected $protected = 'Protected2';
    function printHello() {
        echo $this->public;
        echo $this->protected;
        echo $this->private;
    }
}

$obj2 = new myClass2();
echo $obj->public; // Works
echo $obj2->private; // Undefined
echo $obj2->protected; // Fatal Error
// Shows Public, Protected2, not Private
$obj2->printHello();
?>
```

উপরের ৩৯নং প্রোগ্রামের আউটপুট: Public

১৩.৭ অ্যাবসট্র্যাক এবং ফাইনাল (Abstract Final)

অ্যাবসট্র্যাক কী-ওয়ার্ডটি শুধুমাত্র পিএইচপিতে ক্লাস এবং ক্লাসের ফাংশনের জন্য প্রযোজ্য। যে ক্লাসের ফাংশন অ্যাবসট্র্যাক ঐ ক্লাসটি অবশ্যই অ্যাবসট্র্যাক হতে হবে। তবে অ্যাবসট্র্যাক মডিফায়ারটি ভ্যারিয়েবল এর ক্ষেত্রে ব্যবহার হবে না।

অ্যাবসট্র্যাক ক্লাসকে ঐ ক্লাসের অবজেক্ট দিয়ে ব্যবহার (Access) করা যাবে না। Abstract মডিফায়ারটি ব্যবহার করা হয় যেন ঐ ক্লাসকে ইনস্ট্যানশিয়েটিং (Instantiating) করা না যায়। তবে অ্যাবসট্র্যাক ক্লাসকে ইনহেরিট (Inherit) করতে পারে কোন নন অ্যাবসট্র্যাক ক্লাস। তবে মনে রাখবেন অ্যাবসট্র্যাক ক্লাসের ভিতরের মেথড গুলোর বডি থাকবে না।

```
abstract class cox {
    abstract function hotel ();
}
class bd extends cox {
    function hotel() {
        echo “ Bangladesh ”;
    }
}
$b=new bd
$b->hotel();
```

উপরের প্রোগ্রামটিতে Cox ক্লাসে একটি মেথড হচ্ছে hotel এবং এটি একটি অ্যাবসট্র্যাক মেথড যেহেতু এটি একটি অ্যাবসট্র্যাক মেথড সে জন্য শুধু মেথড এর নামটা আছে। মেথড এর বডি নেই। তাই যে ক্লাস Abstract ক্লাসকে ইনহেরিট (Extends) করে ঐ ক্লাস অ্যাবসট্র্যাক মেথড এর বডিতে কি হবে তা ঠিক করে। অর্থাৎ মেথডটির বডি ডিফাইন করে এই প্রোগ্রামটির আউটপুট হবে Bangladesh ।

> ফাইনাল

এই মডিফায়ার কী-ওয়ার্ডটি ব্যবহার করা হয়, যদি আপনি চান আপনার ক্লাসের রিসোর্স অন্য ক্লাস ব্যবহার (ইনহেরিট এর মাধ্যমে) করবে না। তখন আপনার ক্লাসের পূর্বে Final লিখুন।

এই কী-ওয়ার্ডটি পিএইচপিতে ক্লাস এবং ফাংশন এর ক্ষেত্রে ব্যবহার যোগ্য। তবে ক্লাসের প্রোপার্টি এবং ভ্যারিয়েবল এর জন্য নয়।

Final Class book

```
{
}
```

এখানে book ক্লাসটির কোন সাব ক্লাস থাকতে পারবে না। যেহেতু Final ব্যবহার করা হয়েছে।

১৩.৮ এক নজরে অধ্যায়

> পারলিক ক্লাস এবং পারলিক প্রোপার্টি গুলো যে কোন স্থান থেকে অ্যাকসেস (Access) করা যাবে। প্রাইভেট প্রোপার্টি গুলোকে শুধু ক্লাস এর ভিতরে অ্যাকসেস করা যাবে। প্রোটেক্টেড অনেকটা প্রাইভেট এর মত তবে ক্লাসের সাব ক্লাস প্রোপার্টি গুলো অ্যাকসেস করতে পারবে।

> অ্যাবসট্র্যাক শুধুমাত্র ক্লাস এবং ফাংশন এর ক্ষেত্রে ব্যবহার করা যায়। তবে ফাংশন অ্যাবসট্র্যাক হলে ক্লাসকে অবশ্যই অ্যাবসট্র্যাক হতে হবে। ফাইনাল কী-ওয়ার্ড ব্যবহার করা হয় যেন ঐ ক্লাসের কোন সাব ক্লাস তৈরী করা না যায়।

১৩.৯ প্রশ্ন

১. মডিফায়ার কি?
২. পিএইচপি মডিফায়ার গুলো নাম লিখ।
৩. প্রোটেকটেড মডিফায়ারটি উদাহরণ সহ বর্ণনা কর।
৪. পাবলিক এবং প্রাইভেট মডিফায়ার এর পার্থক্য লিখ।
৫. অ্যাবসট্রাক এবং ফাইনাল কেন ব্যবহার করা হয়?

Quick & Easy Way

Your Professional Trainer...

নিজে নিজে বিজনেস অ্যাপ্লিকেশন তৈরি করুন

অ্যাডভান্স মাইক্রোসফট এক্সেল-২০১৩

এক্সেল বেসিক
ডেটা এন্ট্রি ফর্ম তৈরি
এক্সেল প্রোগ্রাম কালোমাইক্রোসফট
ফর্মুলা এবং ফাংশন
চার্ট এবং গ্রাফিক্স
ম্যাক্রো এবং ডেটাবেজ ম্যানজমেন্ট

১০ টি বাস্তবধর্মী প্রোজেক্ট

বচনা ও সম্পাদনা :
বুকবিডি

More Than Just a Book

বর্তমান টেকনোলজির যুগে আমরা আমাদের দৈনন্দিন কাজ যেমন বাজিঞ্চত ডকুমেন্ট ব্যবসায়িক হিসাব, পরিসংখ্যান এবং লজিক্যাল সবই এক্সেল ব্যবহার করে তৈরি করা সম্ভব তাই খুব সহজে নিজে নিজে এক্সেল শিখার জন্য বুকবিডি সিরিজের এই বইটি ।

With
DVD

Download

Latest Version, 2013, 2010, 2007 www.bookbd.info

অধ্যায়-১৪ কনস্ট্রাকটর এবং ডেস্ট্রাকটর (Constructor & Destructor)

১৪. কনস্ট্রাকটর এবং ডেস্ট্রাকটর (Constructor & Destructor) -----
- ১৪.১ ভূমিকা -----
- ১৪.২ কনস্ট্রাকটর -----
- ১৪.৩ কনস্ট্রাক্টর লিখার নিয়ম -----
- ১৪.৪ কনস্ট্রাক্টর এর প্রকারভেদ (Type of Constructor) -----
- ১৪.৫ ডেস্ট্রাক্টর (Destructor) -----
- ১৪.৬ একনজরে অধ্যায় -----
- ১৪.৭ প্রশ্ন -----

১৪.১ ভূমিকাঃ

সাধারণত কোন ক্লাস এর প্রোপার্টি অ্যাকসেস করতে হলে ঐ ক্লাসের অবজেক্ট ব্যবহার করে করতে হয়। তবে যদি এমন হয় ক্লাসের অবজেক্ট তৈরীর (During Object Creation) সাথে সাথে ক্লাসের কোন প্রোপার্টি নিজে নিজে কাজ করবে এই কনসেপ্টটি হচ্ছে কনস্ট্রাকটর। কনস্ট্রাকটর এবং ডেস্ট্রাকটর (Destructor) হচ্ছে বিশেষ ধরনের ফাংশন/মেথড। কনস্ট্রাকটর রেসপনসিবল অবজেক্ট তৈরীর জন্য কিন্তু ডেস্ট্রাকটর রেসপনসিবল (Responsible) অবজেক্ট ধ্বংস (Destroy) করার জন্য।

১৪.২ কনস্ট্রাকটর (Constructor) কি?

Constructor হচ্ছে বিশেষ ধরনের একটি ফাংশন। ফাংশনকে কল করলে এটি কাজ করে কিন্তু Constructor অটোমেটিক্যালি কল হয়। অর্থাৎ যখন আমরা কোন ক্লাসের অবজেক্ট তৈরি করি তখনই কিন্তু ঐ ক্লাসের Constructor অটো (নিজেই) রান করতে থাকে।

কনস্ট্রাকটর এবং মেথড এর মধ্যে পার্থক্য হচ্ছে অবজেক্ট তৈরির (create) সময় Constructor অটোমেটিক্যালি কল হয় আর সেই অবজেক্ট দিয়ে মেথড অথবা ফাংশনকে কল করতে হয়।

১৪.৩ কনস্ট্রাকটর লিখার নিয়ম

কনস্ট্রাকটর ডিক্লেয়ার করার নিয়ম হচ্ছে প্রথমে function কী-ওয়ার্ডটি লিখতে হবে, তারপর আন্ডারস্কোর () ব্যবহার করবেন এরপর Construct কী-ওয়ার্ডটি লিখবেন এবং তার সাথে দুটো প্রথম ব্রাকেট () ব্যবহার করতে হবে।

নিম্নে দেখানো হচ্ছে কিভাবে Constructor লিখতে হয়

প্রোথাম সিনট্যাক্স ৪ : কনস্ট্রাকটর

```
function_construct() {  
    }  
}
```

অর্থাৎ আমরা কোন কিছু লিখতে চাইলে Constructor বডিতে লিখবো।

```
class com {  
    function_construct() {  
        echo "Dhaka";  
    }  
}
```

সাধারণত com ক্লাসের অবজেক্ট তৈরী করে ঐ অবজেক্ট দিয়ে construct মেথডটিকে কল করার কথা ছিল কিন্তু না। যেহেতু কনস্ট্রাকটর তাকে আলাদা কল করতে হবে না। অবজেক্ট যখন তৈরী হবে তখন নিজে নিজে (Automatically) কল হয়ে যাবে।

```
$cm = new com;
```

আউটপুট প্রদর্শিত হবে Dhaka। এভাবে \$cm -> construct (); লিখতে হবে না।

১৪.৪ কনস্ট্রাকটর এর প্রকারভেদ (Type of Constructor)

সাধারণত তিন ধরনের কনস্ট্রাকটর রয়েছে

১. ডিফল্ট কনস্ট্রাকটর (Default)

২. প্যারামিটারাইজড (Parameterized)

৩. কপি (Copy)

> ডিফল্ট কনস্ট্রাক্টর অর্থাৎ এটি কোন প্যারামিটার নেই (Not Taking any value as a parameter) অর্থাৎ অবজেক্ট তৈরী করার সময় কোন ভ্যালু পাঠানো হয় না। যেমনঃ

```
function_construct ( ) {  
    ↑ এখানে কোন প্যারামিটার নেই।  
}
```

```
Class diu {  
    function_construct() {  
        echo “ Green University ”;  
    }  
}
```

উপরে diu ক্লাসে একটি ডিফল্ট কনস্ট্রাক্টর লিখা আছে এবং ভিতরে ইকো করা আছে। এখন দেখবো কিভাবে অবজেক্ট তৈরীর সাথে সাথে কনস্ট্রাক্টরটি রান করে।

```
$g=new diu();
```

উপরের লাইনটি লিখার সাথে সাথে আউটপুট Green University প্রদর্শন করবে।

```
<?php  
Class k {  
    function_construct() {  
        echo “ Comilla ”;  
    }  
    function_construct() {  
        echo “ Laksham ”;  
    }  
}  
?>
```

এই প্রোগ্রামে অবজেক্ট তৈরীর সাথে সাথে আউটপুট প্রদর্শন করবে Comilla কিন্তু Laksham প্রদর্শন করবে না। যদি করাতে চাই তাহলে অবজেক্ট দিয়ে dhk কে কল করতে হবে।

```
$c=new k();
```

আউটপুট Comilla আসবে।

```
Echo $c-> dhk();
```

আউটপুট Laksham আসবে।

```
<?php  
class BaseClass {  
    public $b=10;
```

অবজেক্ট অরিয়েটেড পিএইচপি

```
function __construct() {
    echo $this->b;
}
function bc() {
    echo "Dhaka";
}
}
$obj = new BaseClass();
?>
```

উপরের প্রোগ্রামের Output: 10

```
<?php
class BaseClass {
    public $b=10;
    function __construct() {
        echo $this->b;
    }
    function bc() {
        echo "Dhaka";
    }
}
$obj = new BaseClass();
$obj->bc();
?>
```

উপরের প্রোগ্রামের Output : 10, Dhaka

```
<?php
class BaseClass {
    public $b;
    function __construct() {
        $this->b=15;
    }
    function bc() {
        echo "Dhaka";
    }
}
$obj = new BaseClass();
$obj->bc();
```

অবজেক্ট অরিয়েটেড পিএইচপি

```
    echo $obj->b;  
?>
```

উপরের প্রোগ্রামের Output: Dhaka, 15

```
<?php  
class BaseClass {  
    public $b=45;  
    function __construct() {  
        $this->b=15;  
    }  
    function bc() {  
        echo "Dhaka";  
    }  
}  
$obj = new BaseClass();  
$obj->bc();  
echo $obj->b;  
?>
```

উপরের প্রোগ্রামের Output: Dhaka, 15

> কনস্ট্রাক্টর এ যখন আমরা প্যারামিটার পাঠাই (Pass) তখন আমরা তাকে প্যারামিটারাইজড কনস্ট্রাক্টর বলে থাকি।

```
Function_construct ($parameter) {  
}
```

এভাবে যতটা ইচ্ছা প্যারামিটার (ভ্যারিয়েবল) কমা দিয়ে লিখা যাবে।

```
Function_construct ($n) {  
}
```

```
Function_construct ($n,$m) {  
}
```

```
class over {  
    Function_construct ($n,$m) {  
        echo “ $name ”;  
    }  
}
```

```
$obj = new over(“ mijan ”);
```

এখানে অবজেক্ট তৈরীর সময় ভ্যালু দিয়েছি mijan যেটি প্যারামিটার হিসেবে \$name এ চলে যাবে এবং প্রোগ্রামটির আউটপুট হবে mijan ।

Constructor কিন্তু ভ্যালু গ্রহণও করে প্যারামিটার হিসেবে। যেমনটি মেথড ভ্যালু গ্রহণ করে। তবে Constructor এ ভ্যালু পাঠাতে হবে অবজেক্ট Create করার সময়।

```
function _Consturct($Name) ← parameter ১টি
{
}
```

```
function _Construct($Name,$Age) ← parameter ২টি
```

✓ মনে রাখবেন Constructor এ একের অধিক প্যারামিটার থাকতে পারে।

প্রোগ্রাম ৩৯৬ (p396.php) একটি ভ্যালু পাঠানো

```
<?php
class human { // { this is starting bracket 1 for this class
    function __construct($name) {
        echo $name;
    }
} // { this is ending bracket 2 for this class human
$akash = new human('Jebin');// call consturctor automatically pass value jebin
// Do not need to call seperately constructor
?>
```

উপরের ৩৯৬ নং প্রোগ্রামের আউটপুট: Jebin

প্রোগ্রাম ৩৯৭ (p937.php) কনস্ট্রাক্টরে একের অধিক ভ্যালু

```
<?php
class human { // this is starting bracket 1 for this class
    function __construct($name, $age) // name and age two parameter value come
    from when create object {
        echo $name;
        echo $age;
    }
} // this is ending bracket 2 for this class human
$akash = new human('Jebin', 21); // call consturctor automatically pass value jebin
// passing two vlaue jebing and 21 during object creation
?>
```

উপরের ৩৯৭ নং প্রোগ্রামের আউটপুট: Jebin21

```
<?php
```

```
class gulshan {
function __construct($n, $m) {
    echo $n;
    echo $m;
}
}
$obj = new gulshan("Mamun", 20);
?>
```

উপরের প্রোগ্রামের Output: Mamun, 20

```
<?php
class gulshan{
function __construct($n, $m, $p) {
    echo $n;
    echo $m;
    echo $p;
}
}
$obj = new gulshan("Mamun", 20, 'k');
?>
```

উপরের প্রোগ্রামের Output: Mamun, 20, k

```
<?php
class gulshan{
function __construct($n, $m, $p) {
    echo $n;
    echo $m;
    echo $p;
}
}
$obj = new gulshan("Mamun", 20, 'k');
?>
```

উপরের প্রোগ্রামের Output: Mamun, 20, k

```
<?php
class gulshan{
    public $n;    public $a;
function __construct($name, $age) {
```

```
        $this->n=$name;
        $this->n=$age;
    }
    function display() {
        echo $this->n; }
    }
    $obj = new gulshan("Mamun", 20);
    $obj->display();
?>
```

উপরের প্রোগ্রামেরঃ Output: 20

```
<?php
class gulshan{
    public $n;    public $a;
    function __construct($name, $age) {
        $this->n=$name;
        $this->n=$age; }
    function display($city) {
        echo $this->n;
        echo $city; }
    }
    $obj = new gulshan("Mamun", 20);
    $obj->display("comilla");
?>
```

উপরের প্রোগ্রামেরঃ Output: 20, comilla

১৪.৫ ডেস্ট্রাক্টর (Destructor)

আমরা যখন কোন অবজেক্টকে destroy (ধ্বংস) করতে চাই তখন ডেস্ট্রাক্টর ব্যবহার করে থাকি। অর্থাৎ কোন ক্লাস এর তৈরি (created) অবজেক্ট যদি আপনি শেষ করে দিতে চান অর্থাৎ ভবিষ্যৎ এ এটি আপনার আর কখনো প্রয়োজন পড়বেনা তখন আপনি Destructor ব্যবহার করে অবজেক্টটিকে destroy করে দিতে পারেন। পিএইচপিতে এটিকে Garbage Collector দিয়ে করা হয়। ডেস্ট্রাক্টরও এক ধরনের মেথড। Constructor এর কাজ হচ্ছে অবজেক্ট তৈরি করা আর Destructor এর কাজ হচ্ছে সেই অবজেক্টকে শেষ করে দেওয়া। অবজেক্ট যখন destroy হয় destructor তখন অটোমেটিক্যালী কল হয়।

প্রোগ্রাম ৩৯৮ (p398.php) অবজেক্ট এবং destroy

```
<?php
```




```
class X{
    public function __construct() // This is constructor {
        echo 'building an object';
    }
    public function __destruct() // destructor to destroy object {
        echo 'destroying the object';
    }
}
$obj = new X(); // when create object both construct and destruct call automatically
?>
```

উপরের ৩৯৮নং প্রোগ্রামের আউটপুট: building an object destroying the object

প্রোগ্রাম ৩৯৯ (p399.php) destroy অবজেক্ট

```
<?php
class Yea {
    public function __destruct() {
        echo 'destroy the object';
    }
}
$obj = new Yea();
// prints Bangladesh
echo 'Banglade4sh ';
?>
```



Banglade4sh destroy the object

চিত্রঃ ১৪.৫ প্রোগ্রাম p399 এর আউটপুট

মনে রাখতে হবে সকল কাজ শেষ করার পর অবজেক্ট destroy করতে হবে অর্থাৎ destruct মেথডটি লিখতে হবে ক্লাসের শেষে যেন তার আগে অন্যান্য কাজগুলো শেষ হয়ে যায়।

প্রোগ্রাম ৪০০ (p400.php) ক্লাস শেষে destruct মেথড

```
<?php
class A {
    public function printMsg() {
        echo 'I still exist';
    }
}
```

```
public function __destruct()    {  
    echo 'destroying the object';  
}  
}  
$b = new A();  
$d = $c = $b;  
$b = null;  
$c->printMsg();  
$d->printMsg();  
?>
```

উপরের 800নং প্রোগ্রামের আউটপুট: I still existI still existdestroying the object

১৪.৬ একনজরে অধ্যায়

কনস্ট্রাক্টর হচ্ছে এক ধরনের বিশেষ মেথড যেটি ক্লাসের মধ্য থাকে এবং যখন অবজেক্ট তৈরী করা হয় তখন কনস্ট্রাক্টর নিজে নিজে (Automatically) রান করে। তবে কনস্ট্রাক্টর লিখতে Function এবং Construct কী-ওয়ার্ড ব্যবহার করা হয়। কিন্তু কনস্ট্রাক্টর লিখার পূর্বে আন্ডার স্কোর (-) ব্যবহার করতে হয়।

মূলত দুই ধরনের কনস্ট্রাক্টর পিএইচপিতে ব্যবহৃত হয়ে থাকে। যেমন ডিফল্ট এবং প্যারামিটারাইজড কনস্ট্রাক্টর। ডিফল্ট কনস্ট্রাক্টর কোন প্যারামিটার থাকে না। কিন্তু প্যারামিটারাইজড কনস্ট্রাক্টর প্যারামিটার হিসেবে কোন ভ্যালু গ্রহন করে না। ক্লাসের অবজেক্ট তৈরীর সময় কনস্ট্রাক্টও এ প্যারামিটার পাঠানো হয়।

প্রোগ্রামিং এ ডেস্ট্রাক্টর মূলত যে কাজটি করে, অবজেক্টকে সমস্ত কিছু থেকে অব্যাহতি দেয় যেমন অবজেক্ট যে মেমরির জায়গা দখল করেছে তা ছেড়ে দিবে। এছাড়াও অবজেক্টটির সাথে যে সমস্ত রিসোর্স গুলোর সম্পৃক্ত (Relation/attach) ছিল, ডেস্ট্রাক্টর মেথড কল করার পর তা আর থাকবে না। তাই তৈরীকৃত অবজেক্টকে ডেস্ট্রয় (Destroy/Delete) করতে চাইলে Destruct ব্যবহার করতে হবে।

১৪.৭ প্রশ্ন

১. কনস্ট্রাক্টর কি?
২. প্রোগ্রামিং এ কেন কনস্ট্রাক্টর ব্যবহার করা হয়?
৩. কনস্ট্রাক্টর এবং ফাংশন এর মধ্য পার্থক্য লিখ।
৪. ডেস্ট্রাক্টর এর কাজ কি।
৫. কনস্ট্রাক্টর এবং ডেস্ট্রাক্টর এর পার্থক্য লিখ।
৬. একটি ক্লাস লিখ। ক্লাসে একটি কনস্ট্রাক্টর আছে যেটি প্রিন্ট করছে Banani। এরপর ঐ ক্লাসের একটি অবজেক্ট তৈরী কর।

বুকবিডি রচিত ও সম্পাদিত অন্যান্য বই সমূহ

- বিগীনিং জুমলা
- অ্যাডভান্সড জুমলা
- প্রফেশনাল জুমলা
- জুমলা টেম্পলেট মেকিং
- বিগীনিং ওয়ার্ডপ্রেস
- অ্যাডভান্সড ওয়ার্ডপ্রেস
- প্রফেশনাল ওয়ার্ডপ্রেস
- ইন্টারনেটে আত্মকর্মসংস্থান-১
- ইন্টারনেটে আত্মকর্মসংস্থান-২
- ই-কমার্স অ্যান্ড জুমলা ভার্টুয়াল
- ই-কমার্স
- ম্যাক্রোমিডিয়া ড্রিমওয়েভার
- সার্চ ইঞ্জিন অপটিমাইজেশন
- অ্যাডভান্সড সার্চ ইঞ্জিন অপটিমাইজেশন
- প্রফেশনাল সার্চ ইঞ্জিন অপটিমাইজেশন
- ফরেক্স ট্রেডিং
- ই-মার্কেটিং
- এইচ টি এম এল-৫
- অ্যাডভান্সড এইচটিএমএল
- পিএইচপি অ্যান্ড মাই এসকিউএল
- অ্যাডভান্সড পিএইচপি অ্যান্ড মাই এসকিউএল
- অবজেক্ট অরিয়েন্টেড পি.এইচ.পি
- ডেটাবেস মাই এসকিউএল
- সি প্রোগ্রামিং
- জাভা প্রোগ্রামিং
- জাভাস্ক্রিপ্ট
- জুমলা টেমপে-ট মেকিং
- অ্যাডভান্সড ফটোশপ
- অ্যাডভান্সড ইলাস্ট্রেটর
- প্রফেশনাল গ্রাফিক্স ডিজাইন
- প্রফেশনার কম্পিউটার নেটওয়ার্কিং
- ওডেস্ক এবং আউটসোর্সিং
- অ্যাফিলিয়েট মার্কেটিং
- ব্লগিং
- সোস্যাল মিডিয়া মার্কেটিং
- ইন্টারনেট মার্কেটিং
- ওয়েব ডিজাইন
- অ্যাডভান্স এক্সেল
- প্রফেশনাল এক্সেল
- অ্যাডভান্স মাইক্রোসফট ওয়ার্ড
- প্রফেশনাল মাইক্রোসফট ওয়ার্ড
- বিগীনিং এক্সেল
- ই-মেইল মার্কেটিং
- মাইক্রোসফট পাওয়ার পয়েন্ট
- মাইক্রোসফট এক্সেস
- প্রোগ্রামিং লজিক সি
- ব্লগিং এন্ড সোস্যাল মিডিয়া
- ওপেন সোর্স সফটওয়্যার
- ক্লাউড কম্পিউটিং
- অ্যানড্রয়েড এবং আই ও এস

“ আমাদের প্রবল ইচ্ছা থাকা সত্ত্বেও আমরা বইটির সম্পূর্ণ কপি আপলোড করতে পারলাম না এজন্য খুবই দুঃখিত। কেননা বইটি প্রিন্ট কপি বাজারে আছে। আপনারা চাইলে বইটি বাংলাদেশের যে কোন লাইব্রেরি থেকে সংগ্রহ করতে পারেন। ”

www.bookbd.info

“ বুকবিডি হচ্ছে বাংলাদেশী প্রফেশনাল বাংলা বই সমূহের ওয়েবসাইট। যেখান থেকে আপনি ই-বুক বিনা মূল্যে ডাউনলোড করতে পারবেন। যে বইগুলো আপনাকে আইটি আউটসোর্সিং- এবং আইসিটিতে প্রফেশনাল ক্যারিয়ার গড়ে তুলতে সাহায্য করবে। এ ছাড়াও নিজে নিজে কোন প্রকার ট্রেনিং ছাড়াই যে কোন বিষয়ের উপর প্রফেশনাল দক্ষতা অর্জন করতে পারেন এই বইগুলো পড়ে। আর আপনাদের কোন পরামর্শ থাকলে অবশ্যই আমাদেরকে জানাবেন। ”

এই ঠিকানায়ঃ- infobook7@gmail.com.