**Classes and Objects**

A class is a template for objects, and an object is an instance of class.

In a class, variables are called properties and functions are called methods!

Objects of a class are created using the new keyword.

Class is a programmer-defined data type.

**Property**

ক্লাস এর মেম্বার ভ্যারিয়েবল।

**Member function and veriable**

ক্লাসের ভিতর ভেরিয়েবলকে বলা হয় প্রোপর্টিজ বা মেম্বার ভেরিয়েবল এবং ফাংশনকে বলা হয় মেথড বা মেম্বার ফাংশন বা (behaviour)বিহেবিয়ার।

**Constructor**

পিএইচপির বিশেষ ধরণের বিল্ট-ইন মেথড। অবজেক্ট তৈরির সময় প্রোপার্টির ভ্যালু এসাইনে সম্মতি দেয়। অবজেক্ট তৈরি হলে এই মেথডটি স্বয়ংক্রিয়ভাবেই সম্পাদিত হয়। দুটি আন্ডারস্কোর(\_\_) দিয়ে construct মেথড শুরু হয়।

**Function**এবং **Constructor** এর পার্থক্যগুলো

* Function হল Group of Code যা একটি নির্দিষ্ট কাজ করে অন্যদিকে Constructor এর কাজ হল class এর Instance তৈরি করা।
* Function এর return type থাকে কিন্তু Constructor এর কোন return type নেই এমনকি void ও ব্যবহার করা যায় না ।
* Function এ final, static, abstract এই ধরনের Modifier গুলো ব্যবহার হয় কিন্তু Constructor এ এই ধরনের Modifier গুলো ব্যবহার করা যায় না।
* Function এ this, super কিওয়ার্ড ব্যবহার হয় Working Class এবং Parent Class এর Instance কে Refer করার জন্য অন্যদিকে constructor এ this, super কিওয়ার্ড ব্যবহার হয় Working Class এবং Parent Class এর Constructor কে Call করার জন্য।

**PHP: Overloading and Overriding –**

**অভারলোডিং** তখনই ঘটে যখন একই ক্লাসে দুই বা তার অধিক মেথড এর নাম একই কিন্তু প্যারামিটার ভিন্ন হয়।

**Overriding** এর অর্থ হলো দুটি মেথডের নাম এবং প্যারামিটার(method signature) একই। একটি মেথড থাকে প্যারেন্ট ক্লাসে এবং অপরটি থাকে চাইল্ড ক্লাসে।

Main parts of oop :- (AEIP)

1. **Abstraction**:-

* An abstract class is a class that contains at least one abstract method. An abstract method is a method that is declared, but not implemented in the code.
* An abstract class or method is defined with the abstract keyword:
* **Abstract Class কি?**

Abstract Class এমন এক ধরনের ক্লাস যার অবজেক্ট সৃষ্টি করা যায় না। অন্য ক্লাস(subclass) থেকে extend করাই হল এর উদ্দেশ্য।

* **Abstract Method কি?**

Abstract method হল এমন এক ধরনের মেথড যার শুধু ঘোষণা (declaration) থাকে,বাস্তবায়ন(implementation) থাকে না।

Abstract Method কেন ব্যবহার করা হয়?

ধরি, আমরা একটি মানুষের আচরনের ক্লাস তৈরি করতে চাই, যার Base Class শুরু হবে মানুষ নাম দিয়ে। বিভিন্ন মানুষের বিভিন্ন রকম আচরন থাকতে পারে যেমন কেউ গান পছন্দ করে, কেউ নাচ পছন্দ করে, কেউ ক্রিকেট খেলা পছন্দ করে কিন্তু সব মানুষের কমন কিছু

বৈশিষ্ট্য আছে যেমন সব মানুষেই খাবার গ্রহন করে, সব মানুষেই ঘুমায়। যখন একটি বেস ক্লাসে ভিন্ন ভিন্ন বৈশিষ্ট্যর মেথড থাকে,তখন ওই ক্লাসকে abstract class হিসেবে ঘোষণা করা হয়।

**Interface :**

* ইন্টারফেসের মধ্যে কোন property থাকতে পারবেনা। Interface এর method অবশ্যই public হতে হবে। তবে Interface এর সকল method ই abstract এর মত। function এর মধ্যে কোন implement করা যায়না। Interface কে একটি class implement করতে হলে implement keyword ব্যবহার করতে হয়।
* আমরা একটি চাইল্ড ক্লাসে অসংখ্য ইন্টারফেস সংযুক্ত করতে পারি। একটি ক্লাসে একাধিক ইন্টারফেস সংযুক্ত করার জন্য কমা(,) চিহ্ন ব্যবহার করে ইন্টারফেসসমূহক পৃথক করা হয়।
* Interface class কে অ্ন্য ক্লাস implement করতে হলে তার সকল মেথড কে implement কেরতে হয়।
* ইন্টারফেসে ক্লাসের সকল মেথড অবশ্যই public হতে হবে, কিন্তু অ্যাবস্ট্রাক্ট ক্লাস সকল অ্যাক্সেস মডিফায়ার সমর্থন করে।
* একটি চাইল্ড ক্লাস একাসাথে একাধিক ইন্টারফেস ব্যবহার করতে পারে, কিন্তু অ্যাবস্ট্রাক্ট ক্লাসের ক্ষেত্রে চাইল্ড ক্লাস একের অধিক অ্যাবস্ট্রাক্ট ক্লাস থেকে ইনহেরিট হতে পারে না।

|  |  |  |
| --- | --- | --- |
| ইন্টারফেস এবং অ্যাবস্ট্রাক্ট ক্লাসের মধ্যে পার্থক্য | | |
|  | **ইন্টারফেস** | **অ্যাবস্ট্রাক্ট ক্লাস** |
| কোড | অ্যাবস্ট্রাক্ট মেথড, কনস্ট্যান্ট | অ্যাবস্ট্রাক্ট মেথড, কনস্ট্যান্ট, কনক্রিট মেথড, কনক্রিট ভ্যারিয়েবল |
| অ্যাক্সেস মডিফায়ার | public | public private protected |
| প্যারেন্ট ক্লাসের সংখ্যা | একটি চাইল্ড ক্লাসের একাধিক প্যারেন্ট ক্লাস বা ইন্টারফেস থাকতে পারে। | একটি চাইল্ড ক্লাসের শুধুমাত্র একটি প্যারেন্ট ক্লাস বা অ্যাবস্ট্রাক্ট থাকবে। |

**Syntax of interface :**

Interface interfaceName{

body—----

}

Interface car{

Public function sound();

}

## **Traits :** PHP তে multiple inheritance সাপোর্ট করেনা। এই সমস্যা solve করার জন্য Traits use করা হয়। Traits এর কোন object/instance create করা যায়না। একটি class multiple traits কে use করতে পারে।

**Syntax of traits:**

Traits traitsName{

body —--------------

}

Traits message{

Public function show(){

—---------

}

}

How to use traits—

Class এর মধ্যে traits কে use করার জন্য use keyword ব্যবহার করা হয়।

Example– use traitsName

Class a {

Use message;

}