Lesson:

Introduction to OOPs

PW SKILLS

Pre-Requisites

- Fundamentals of Object-Oriented Programming Language
- Class and object

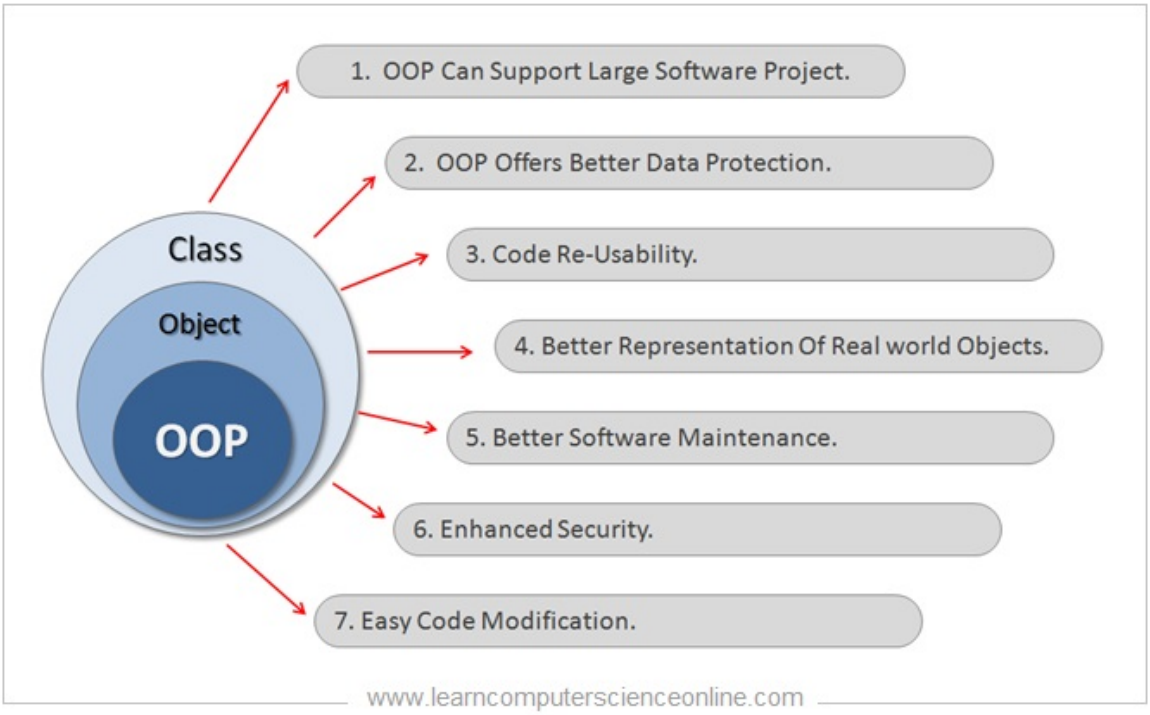
Programming is a wonderful tool that allows us to create apps and software, build websites and online shops, publish e-books and much much more. In the early era of programming, the concepts that were used put a lot of pressure on the developer to model solutions in a certain way, which in turn, made this domain inaccessible to most of the people. Learning the concepts was tough in those days and applying it to solve real-world problems was even more challenging. As we progressed, the programming languages were developed to support the concepts that modeled problem scenarios in the easiest way possible. Java, for instance, is known for its closeness to real-world perception.
This brings us to the concept of Object Oriented Programming approach.

Topic : Object-Oriented Programming Concept(OOP)

Object-oriented programming (OOP) is a computer programming model that organizes software design around data, or objects, rather than functions and logic.
OOP focuses on the objects that developers want to manipulate rather than the logic required to manipulate them. This approach to programming is well-suited for programs that are large, complex and actively updated or maintained.

Object oriented approach is said to have been the most successful in handling real world problems.

The diagram below pretty much sums up the benefits that OOP brings to the table



OOP concept works on the principle of considering everything around as a class and object which has some properties and behavior, which is true for almost everything in the world. You might be wondering now, what are these classes, objects, properties and behavior ? Let us understand it step by step .

Topic: What are Classes and Objects?

The fundamental elements of OOP are classes and objects. Once we understand these and the way they are used and perceived, we will be equipped with half of the solution-building skills required for any given problem.

Let's look at them one by one.

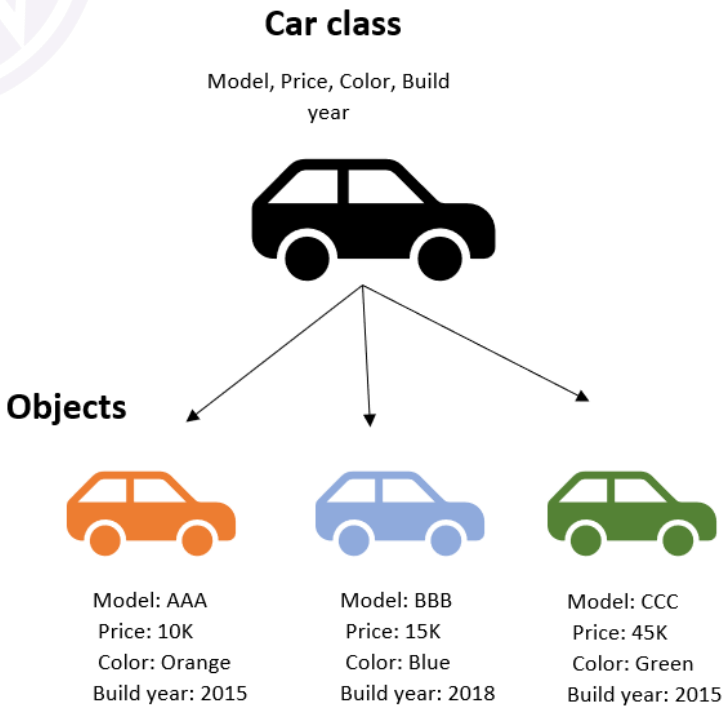
A Class is a user-defined data-type which has data members and member functions. It is a logical entity. A class can also be understood as a group of objects that have common properties. It is a template or blueprint from which objects are created.

Confusing ?! Class and objects are so interlinked that they will have to be seen together to get a clear picture. But before that let us look at what objects are.

An Object is an identifiable entity with some characteristics and behaviour. **It** represents a real-life entity. It is the most basic unit of object-oriented programming. Object is also identified as an instance of a Class which means that when a class is defined, no memory is allocated but when it is instantiated (i.e. an object is created) memory is allocated.

Objects are the things that we first think about while designing a program and they are also the units of code that are eventually derived from the process. All of this might sound too much of an information to you in the first go. But as we move ahead, all of this will become a routine thing which comes naturally to coders like you !

Look at the example below :



Here, imagine, Car as a class, with main attributes(identifying elements) such as model, price, color, and build year. We can create as many objects as we want from the Car class. All of these would be Cars but with different attributes such as model, price, color and build year.
Let us look at another similar example:

Class: Vehicle
Objects : Car,truck,bike.

Here, as you can see, class is a template for objects, and an object is an instance of a class.
In this example, Vehicle is a class and Car, truck and bike are objects.

Now that we have understood classes and objects and how closely linked they are, we will now move ahead to know them better.

Object has three characteristics -

Identity - Unique name of the object
State - Attributes of an object
Behavior - methods or logic executed on the object

For Example - car

Identity - Name of car or model no
State - Color, Engine, etc
Behavior - car speed, etc

Characteristics of a class:

Class consists of attributes or methods which can be executed by objects(instance of the class)
Components of a class : A class needs to have all of these for its existence.

- Access Modifiers - scope of class where it can be used
- class keyword - used to create a class
- Class name - Name of class
- Body - The body of class which consists of attributes, objects, methods and so on

Lets now look at some examples for better understanding.

1. Create a class named "Person" having a variable age:

```
public class Person{
  int age=20;
}6p
```

Explanation- Here, public is access specifier(which defines the limit to which a class can be accessed, it is explained in next lecture),class is the keyword that is used to create class and Person is the name of class and the Body contains a variable age of int type.

Create an Object

In Java, an object is created from a class. We have already created the class named Person, so we can use this class to create objects.

To create an object of Person class, specify the class name, followed by the object name, and use the keyword new:

```
ClassName object = new ClassName();
```

Let us see how it would appear in a code :

Create an object of Person class

```
public class Person {
  int age = 20;
 public static void main(String[] args) {
    Person obj1 = new Person();
    System.out.println(obj1.age);
  }
}
```

```
public class Person {
int age = 20 ;
public static void main (String[]args){
Person obj1 = new Person();
System.out .println(obj1.age);
}
}
```

```
Output: 20
```

Object and Class Example:

In the example discussed ahead, we have created a Student class which has two data members rollNo and studentName. We are creating the object of the Student class using 'new' keyword and printing the object's value.

Here, we are creating a main() method inside the class.

3. Java Program to illustrate how to define a class and fields

```
        //Defining a Student class.
public class Student{
 //defining fields
 int rollNo; //field or data member or instance variable
String studentName;
 //creating main method inside the Student class
 public static void main(String args[]){
  //Creating an object or instance
  Student obj1=new Student(); //creating an object of Student
  //Printing values of the object
  System.out.println(obj1.rollNo);//accessing member through reference variable
  System.out.println(obj1.studentName);
 }
}
```

```
Output: 0
             null
```

Object and Class Example: main outside the class

In real time development, we create classes and use from another class. It is a better approach than the previous one because this has the advantage that we can reuse our .class file somewhere in other projects without compiling the code again.

Let's see an example, where we are having the main() method in another class.

We can have multiple classes in different Java files or in one Java file. If we define multiple classes in one Java source file, it is preferred to save the file name with the class name which has the main() method with the public access modifier.

Wondering why ?

Let us understand the reason behind this...

- To inform the JVM that this is an entry point ,filename must have the same name as the public class name in that file.
- Suppose, we create a program which has more than one class. After compiling the java source file, it will generate as many number of the .class files as the classes in the program. In this condition, we will not be able to easily identify which class needs to be interpreted by the java interpreter and which class contains the Entry point for the program.

So, for this reason it is always preferred to save the file name with the class name which has the main() method with the public access modifier. We will always follow this in our programs.

Let us go through some questions to have a clarity on OOP concepts learnt so far.

4. Java Program to demonstrate having the main method in another class

```
//Creating Student class.
public class Student{
    int rollNo;
    String studentName;
}
//Creating another class NewStudent1 which contains the main method
class NewStudent1{
  public static void main(String args[]){
  Student obj1=new Student();
  System.out.println(obj1.rollNo);
  System.out.println(obj1.studentName);
 }
}
```

Output: 0
null

5. We can also create multiple objects of the same class and information can be stored in it through reference variables.

For example:

```
class Student{
     int rollNo;
     String studentName;
}
class Student2{
public static void main(String args[]){
  //Creating objects
  Student obj1=new Student();
  Student obj2=new Student();
  //Initializing objects
  obj1.rollNo=100;
  obj1.StudentName="Ram";
  obj2.rollNo=200;
  obj2.StudentName="Shyam";
  //Printing data
  System.out.println(obj1.rollNo+" "+obj1.StudentName);
  System.out.println(obj2.rollNo+" "+obj2.StudentName);
 }
}
```

```
Output: 100 Ram
             200 Shyam
```

Some advantages of Object-oriented programming are :

- OOP is faster and easier to execute
- OOP provides a clear and simple structure for the programs
- OOP makes it possible to create fully reusable applications with less code and shorter development time.

Let us now look at a few MCQs to test our knowledge.

MCQs:

Q1. Which of these operators is used to allocate memory for an object in java?

a) malloc
b) alloc
c)new
d)realloc

Ans: c)new

Explanation: The operator "new" dynamically allocates memory for an object and returns a reference to it. This reference is the memory address of the object allocated to "new".

Q2. Select the incorrect statement:

a) OOPS refers to using objects in programming
b) A class is a user defined blueprint from which objects are created.
c) Objects of the same class have different properties.
d) Object is an instance of a class.

Ans: c) Objects of the same class have different properties.

Explanation: All of the objects created from the same class possess the class's properties.

Q3. Which of the following methods can be used to create an object of player class?

a) Player p1=new Player;
b) Player p1;
c) Player p1=new Player();
d)Player p1=" ";

Ans: c) Player p1=new Player();

Q4. Which line of the following code would give an error?

```
 class Student{
    int rollNo;
    String name;
}
class Demo {
    public static void main (String[] args) {
        Student s=new Student();        //Line 1
        rollNo=10;                        //Line 2
        s.name="Ram"                       //Line 3
    }
}
```

a) Line 1
b) Line 2
c) Line 3
d) No error

Ans: b) Line 2

Explanation: In line 2 , rollNo is not a variable in main. So, we need to use the object of class Student.

Q5. Creating an object from a class is known as ?

A) Instantiating
B) Initializing
C) Interfacing
D) None of the above

Ans: a) Instantiating

Explanation: In Java, instantiation means to call the constructor of a class that creates an instance or object of the type of that class

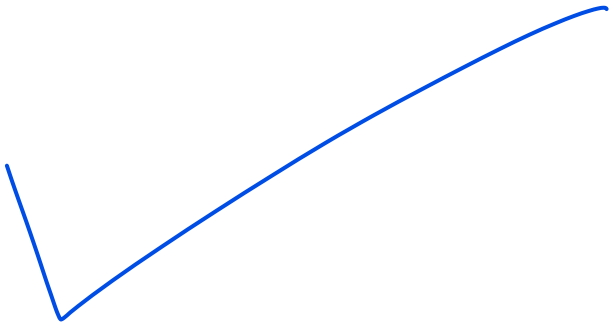
That is all for OOP wrt Java implementations that we would be doing in the forthcoming lectures.

Follow the course for more !

Till then, happy learning !!

Upcoming Class Teasers:

- Java Methods
- Types Of Methods: User defined and Standard Library



21/08/2023