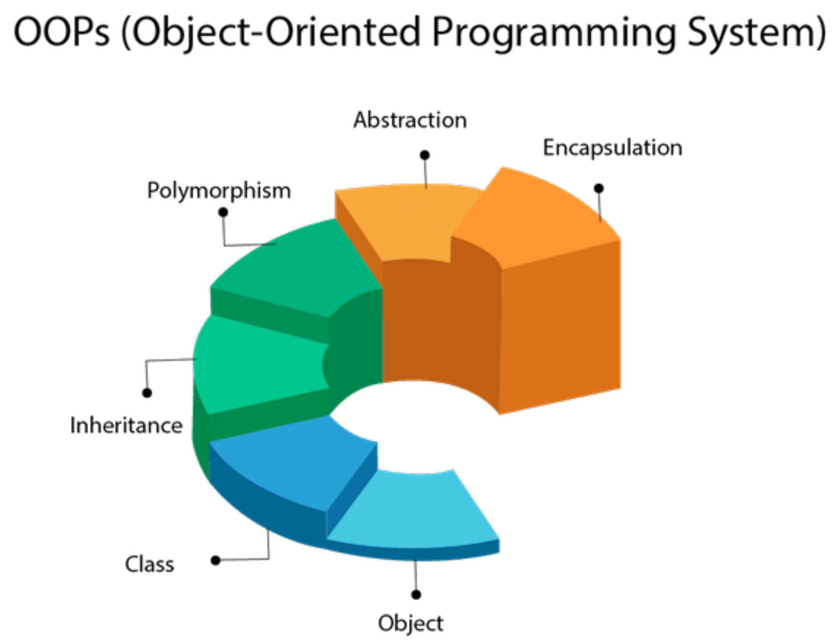
Object Oriented Analysis and Design using Java

Prof: Mahitha G

Department of Computer Science and Engineering

Object Oriented Analysis and Design using Java

Introduction to OO Programming



Pfro: Mahitha G

Department of Computer Science and Engineering

Object Oriented Analysis and Design using Java

Unit-01 : Object Oriented Programming

~~**1:** Introduction to course, Introduction to object-oriented concepts, Object Based Programming: JVM~~

~~**2:** Abstraction, Encapsulation, Composition~~

3: ~~**Class Attributes, Behaviour, Objects, and Methods**~~

4: Interface and Implementation: Role of Constructors and Destructors, Garbage Collector

5: Parameter Passing, Value Type and Reference Type

6: Overloading of Methods Model

7: Java Recursion

8: Class Attributes and Behaviour: Difference between Class Methods and Instance Methods

9: Inheritance: Concepts of Single Rooted Hierarchy and Interface

10: Abstract Class in Programming Languages, Object Class in Java

Object Oriented Programming: Reference

T1 : Chapter 6: Introducing Classes

1: “Java the Complete Reference”, Herbert Schildt ,McGraw-Hill ,11th Edition, 2018.

Constructor

- A constructor **initializes an object** when it is created.
- It has the **same name as its class** and is syntactically similar to a method.
- Constructors have **no explicit return type**.
- Typically, you will use a constructor to give initial values to the instance variables defined by the class, or to perform any other **start-up procedures** required to create a fully formed object.
- All classes have constructors, whether you define one or not, because Java automatically provides a **default constructor** that initializes all member variables to zero or corresponding default value. However, once you define your own constructor, the default constructor is no longer added.
- Each time a object is created using new operator, constructor is invoked to **assign initial values to the data members of the class.**

Object Oriented Programming: Constructor

```
Class Student
{
Student( )
{
// initialization
  }
}
```

Next we create an object of the above class.

```
Student obj = new Student( );
```

Object Oriented Analysis and Design using Java

Object Oriented Programming: Types of Constructors

Default constructor :

- A constructor that has no parameters. If we don’t define a constructor for a class, then compiler creates a default constructor.
- Default constructor provides default values to the objects like 0, false, null etc depending on the data type of the instance variables.

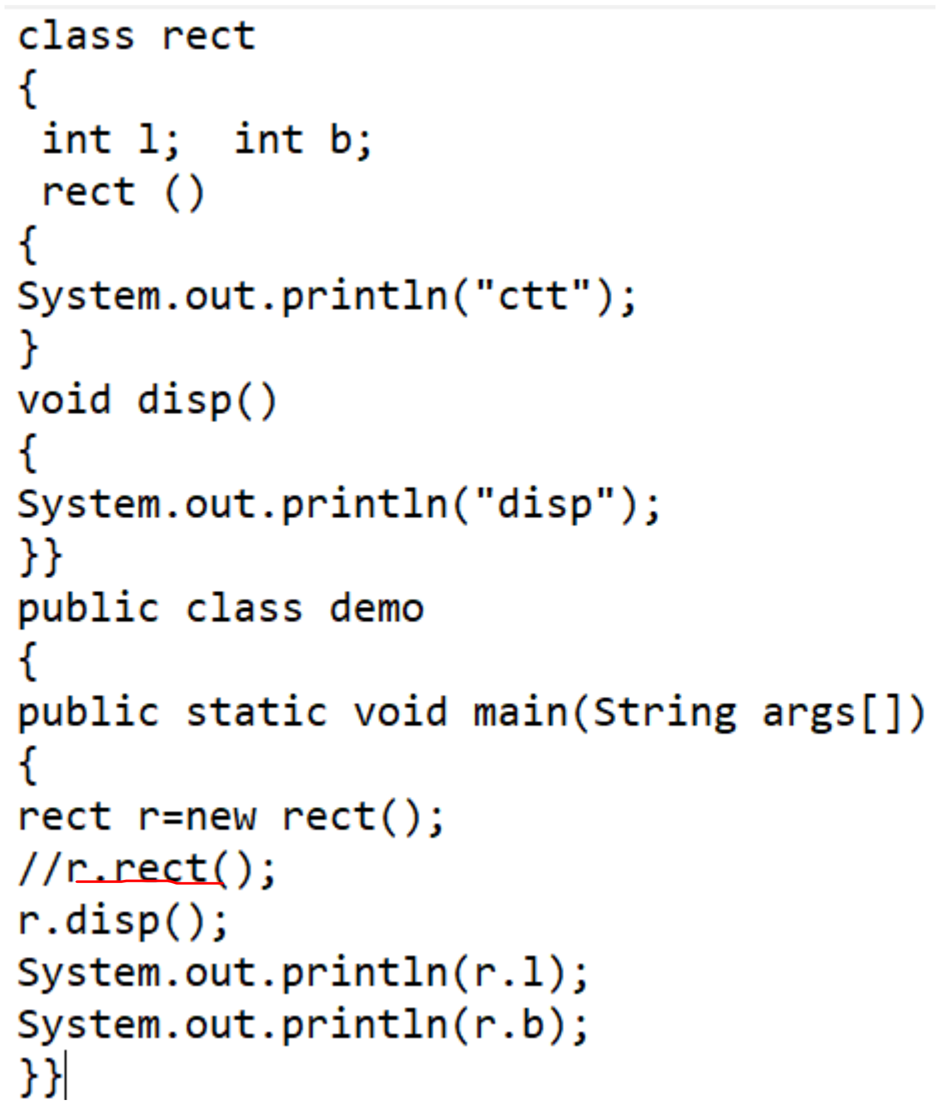
Parameterized constructor:

- A constructor with parameters.
- To initialize the fields of a object with given values
- There are no return value statements in a constructor but constructors return the current class instance.

//program for demonstration

Object Oriented Analysis and Design using Java

Object Oriented Programming: Access Modifiers - Example



```
class rect
{
 int l;  int b;
 rect ()
{
System.out.println("ctt");
}
void disp()
{
System.out.println("disp");
}}
public class demo
{
public static void main(String args[])
{
rect r=new rect();
//r.rect();
r.disp();
System.out.println(r.l);
System.out.println(r.b);
}}
```

Object Oriented Programming: Access Modifiers - Example

If you make any class constructor private, you cannot create the instance of that class from outside the class.

By default the access modifier is “default”

```
class A{
private A()  { }                                    //private constructor
void msg(){System.out.println(“Welcome to OOAD with java class");;}
}
public class Sample
{
 public static void main(String args[]){
   A obj=new A();                                   //Compile Time Error
 }
}
```

Object Oriented Programming: Garbage Collector

- Java Garbage Collection is the process to identify and remove the unused objects from the memory and free space.
- One of the best feature of java programming language is the **automatic garbage collection**, unlike other programming languages such as C where memory allocation and de-allocation is a manual process.
- **Garbage Collector** is the program running in the background that looks into all the objects in the memory and find out objects that are not referenced by any part of the program.
- All these unreferenced objects are deleted and space is reclaimed for allocation to other objects.

Object Oriented Analysis and Design using Java

Object Oriented Programming: finalization

There are certain actions to be performed before an object is destroyed like:

- Closing all the database connections or files
- Releasing all the network resources
- Other Housekeeping tasks
- Recovering the heap space allocated during the lifetime of an object
- Release of release locks

Java provides a mechanism called finalization to do this through finalize( ) method.

Object Oriented Programming: finalize ( )

General form of finalize ( ) method

```
protected void finalize( )
{
//finalization code here
//specify those actions that must be performed before an object is destroyed.
}
```

- Java run time calls this method whenever it is about to recycle an object of the class.
- Keyword protected is used to prevent access to finalize ( ) by the code defined outside its class.
- Called just prior to garbage collection and not called when an object goes out of scope

PES UNIVERSITY ONLINE

THANK YOU

Mahitha G
Department of Computer Science and Engineering
mahithag@pes.edu