Object Oriented Programming using Java – Part 1

By

Dr. Sunirmal Khatua,

Visvesvaraya Young Faculty Fellow, Govt. of India

Assistant Professor, University of Calcutta

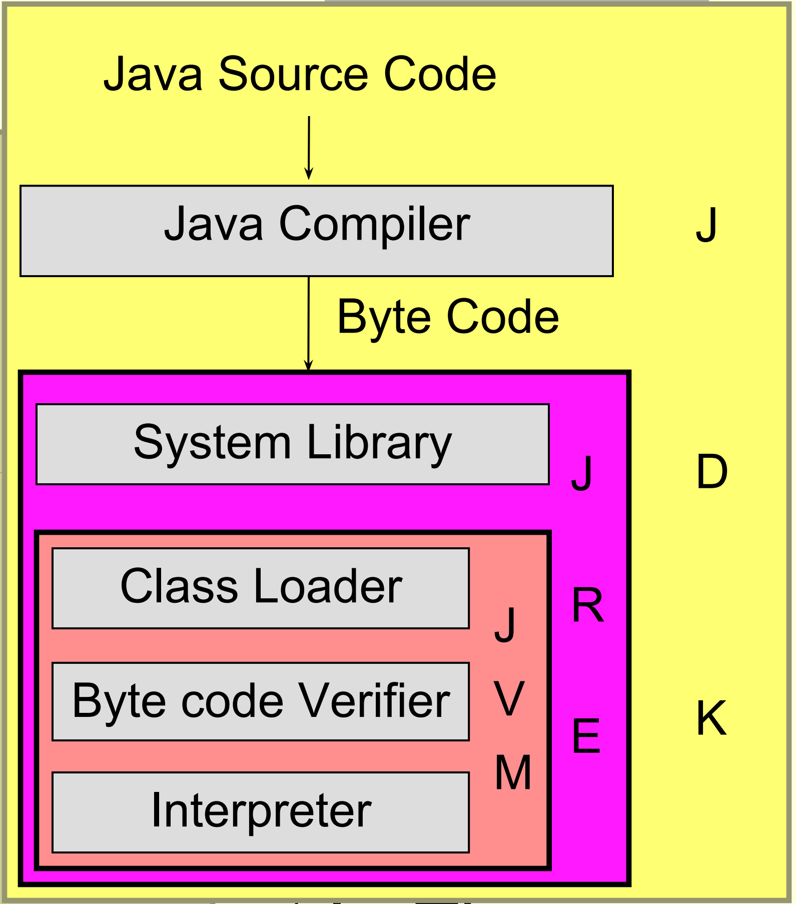
Features of Java

- Object Oriented

```
class C{

    int i ;

    void setI(int i1){
        i = i1;
    }

    int getI(){
        return i;
    }

}

C  ob1  =  new C();
ob1.setI(10);

C  ob2  =  new C();
ob1.setI(20);
```

State

Behavior



ust In-Time

Hello World in Java

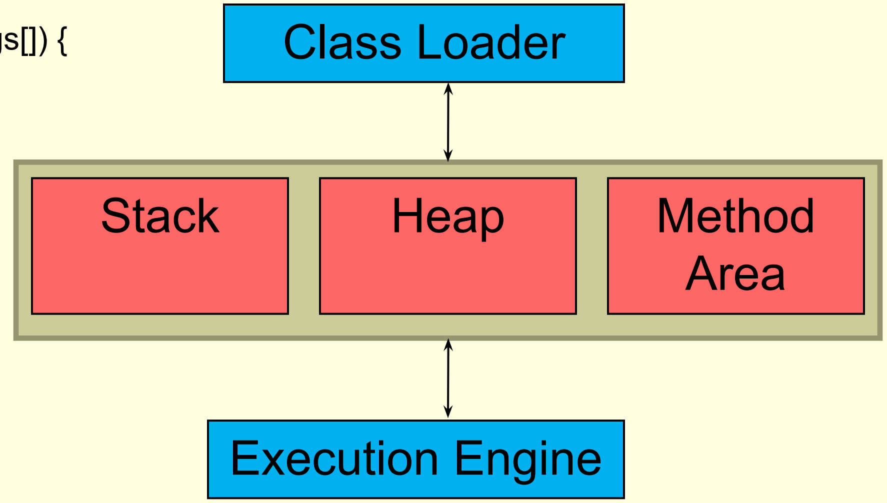
```
class HelloWorld {

     public static  void  main(String args[ ]) {

                    System.out.println(“Hello World……”);

      }

}
```

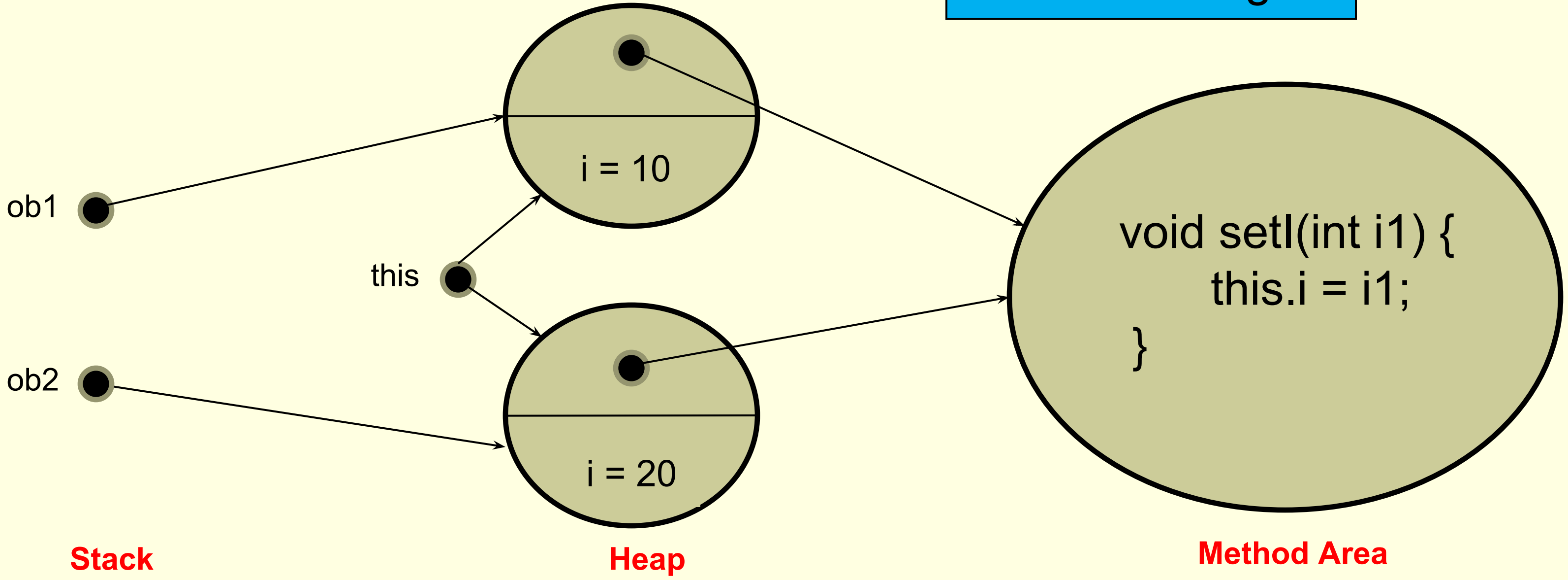
Note : You have to save the program with the same name as the name of the class containing the main() method

In-Memory Structure of JRE

```
class C {
    int i;
    void setI(int i1){
          i = i1;
     }
}
```

```
class  Test {
    public static void main(String args[]) {
          C ob1 = new C();
          ob1.setI(10);
          C ob2 = new C();
          ob2.setI(20);
    }
}
```





Method Overloading

- Used to allow same name for two different methods within the same class.
- The overloaded methods must differ in signature.
- Signature is identified by:
  - *No. of arguments*
  - *Types of arguments*
  - *Order of arguments*
- It implements compile time polymorphism

Constructor

- A special method having the following properties:
  - Same name as the class
  - Doesn’t have any return type
  - Used to initialize the object
  - Every class must have a constructor. If we missed one Java environment will provide a default constructor automatically.
- Types of Constructor
  - Default Constructor
  - Parameterized Constructor
  - Copy Constructor

Constructor(cont.)

- We can call the overloaded constructor by using the keyword “*this*”
- this (if present) must be the first statement within the constructor of a class

Finalize

- A special method which is called before removing the object from memory.

- It has the signature:

```
protected void finalize(){
      // finalize code
}
```

Garbage Collector(GC)

- A demon thread running inside the JRE used to free the memory of unused objects.
- Most of the times GC sleeps and at regular interval it checks for garbage collection.
- Garbage collection can’t be forced but can be requested using *System.gc()*

Lifecycle of an Object

- Created
- In Use
- Unreachable
- Collected
- Finalized
- Deallocated

1. Space is allocated
2. Super class constructor is called
3. Instance variables are initialized
4. Constructor is executed

Static Modifier

- A static member is not tied to any instance and is stored in the Class Data of the Method Area
- A static member can be accessed from the context of the class as well as objects
- A non-static method can access both static or non-static member.
- A static method can access only static members

Static Block

- A block of code inside a class and outside of any method having the signature:

```
static{
   //static initialization
      }
```

- Used to initialize during loading of a Class