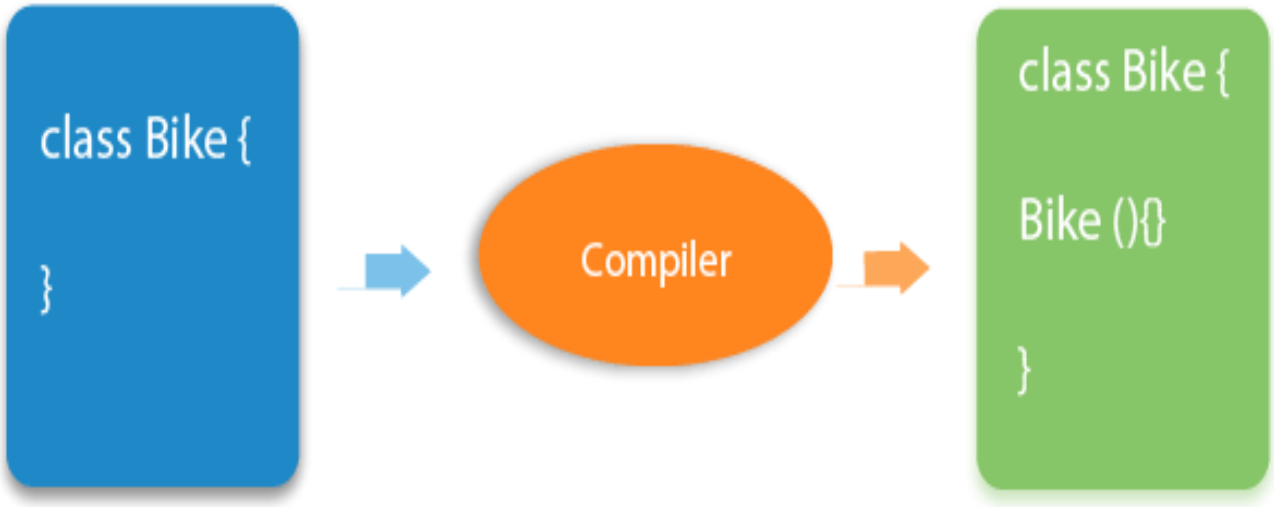
B.Sc (Phy Sc.)
CS

Programming in JAVA

Constructor of class

- Every time an object is created using the new keyword, at least one constructor is called.
- if there is no constructor available in the class. In such case, Java compiler provides a default constructor by default
- It is called constructor because it constructs the values at the time of object creation.
- NOTE: It is not necessary to write a constructor for a class. It is because java compiler creates a default constructor if your class doesn't have any.


class Bike {
}
Compiler
class Bike {
Bike (){}
}

rules defined for the constructor.

- Constructor name must be the same as its class name
- A Constructor must have no explicit return type
- A Java constructor cannot be abstract, static, final

Types of Java constructors

There are two types of constructors in Java:

- Default constructor (no-arg constructor)
- Parameterized constructor

```
class Demo{
//creating a default constructor
Demo()
{System.out.println(“Demo  is created");}
//main method
public static void main(String args[]){
//calling a default constructor
Demo b=new Demo();
}
}
```

```
//Let us see another example of default constructor
//which displays the default values
class Student3{
int id;
String name;
//method to display the value of id and name
void display()
{System.out.println(id+" "+name);}
public static void main(String args[]){
//creating objects
Student3 s1=new Student3();
Student3 s2=new Student3();
//displaying values of the object
s1.display();
s2.display();
}  }
```

Parameterized Constructor

- A constructor which has a specific number of parameters is called a parameterized constructor.

Why use the parameterized constructor?

- The parameterized constructor is used to provide different values to distinct objects.

```
class Student{
   int id;
   String name;
   //creating a parameterized constructor
   Student(int i,String n){
   id = i;
   name = n;
   }

   //method to display the values
   void display(){System.out.println(id+" "+name);}

   public static void main(String args[]){
   //creating objects and passing values
   Student s1 = new Student(111,"Karan");
   Student s2 = new Student(222,"Aryan");
   //calling method to display the values of object
   s1.display();
   s2.display();
  }  }
```

Constructor overloading

- Constructor overloading in Java is a technique of having more than one constructor with different parameter lists. They are arranged in a way that each constructor performs a different task. They are differentiated by the compiler by the number of parameters in the list and their types