1)

```
class Main {
	private String name;
	//constructor
	Main()
	{
		System.out.println("constructor is called");
	    name="programiz";
	}
		public static void main(String[] args) {
			// TODO Auto-generated method stub
			//constructor is invoked while
			//creating an object of the main class
			Main obj=new Main();
			System.out.println("The name is= "+obj.name);
	}

}
```

output:

```
constructor is called
The name is= programiz
```

****

2)

```
public class Main1 {
	int i;
	//constructor with no parameter
	private Main1()
```

```
	{
	i=5;
	System.out.println("constructor is called");
	}
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		//calling the constructor without any parameter
		Main1 obj=new Main1();
		System.out.println("The value i is= "+obj.i);
	}

}
```

output:

```
constructor is called
The value i is= 5
```

```
********************************************************************************
3)
class Company {
	String name;
	//public constructor
	public Company()
	{
		name="programiz";
	}
}
class Main2
{
```

```
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		//object is created in another class
		Company obj=new Company();
		System.out.println("The company name is= "+obj.name);

	}

}
```

output:

```
The company name is= programiz
```

****

4)

```
class Main3 {
	String languages;
	//constructor accepting single value
	Main3(String lang)
	{
		languages=lang;
		System.out.println(languages +" programing language ");
	}
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		//call constructor by passing a single value
		Main3 obj1=new Main3("java");
		Main3 obj2=new Main3("python");
		Main3 obj3=new Main3("c");
```

```
	}

}
```

output:

```
java programing language
python programing language
c programing language
```

```
*********************************************************************************
```

5)

```
class Main4 {
	int a;
	boolean b;
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		//a default constructor is called
		Main4 obj=new Main4();
		System.out.println("Default Value: ");
		System.out.println("a= "+obj.a);
		System.out.println("b= "+obj.b);
	}

}
```

output:

```
Default Value:
a= 0
b= false
```

```
*********************************************************************************
```

6)

```
class Main5 {
	int a;
	boolean b;
	//a private constructor
	private Main5()
	{
		a=0;
		b=false;
	}
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		//call the constructor
		Main5 obj=new Main5();
		System.out.println("Default Value: ");
		System.out.println("a= "+obj.a);
		System.out.println("b= "+obj.b);
	}

}
```

output:

```
Default Value:
a= 0
b= false
```

****

7)

```
public class Main6 {
	String language;
	//constructor with no parameter
	Main6()
	{
		this.language="java";
	}
	//constructor with single parameter
	Main6(String language)
	{
		this.language= language;
	}
	public void getName()
	{
		System.out.println("programing languge: "+ this.language);
	}
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		//call constructor with no parameter
		Main6 obj=new Main6();
		//call constructor with single parameter
		Main6 obj1=new Main6("python");
		obj.getName();
		obj1.getName();
	}

}
```

output:

```
programing languge: java

programing languge: python

********************************************************************************
```

1)

```
//Java Program to create and call a default constructor
class Bike1{
//creating a default constructor
	Bike1()
	{
	System.out.println("Bike is created");
	}
	//main method
	public static void main(String args[]){
		//calling a default constructor
		Bike1 b=new Bike1();
	}
}
```

output:

```
Bike is created

********************************************************************************
```

2)

```
//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();
      }
}
```

output:

```
0 null
0 null
```

****

3)

```
//Java Program to demonstrate the use of the parameterized constructor.
class Student4{
    int id;
    String name;
    //creating a parameterized constructor
    Student4(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
    Student4 s1 = new Student4(111,"Karan");
    Student4 s2 = new Student4(222,"Aryan");
    //calling method to display the values of object
    s1.display();
    s2.display();
   }
}
```

output:

```
111 Karan
222 Aryan
```

****

4)

```
//Java program to overload constructors
class Student5{
    int id;
    String name;
    int age;
    //creating two arg constructor
    Student5(int i,String n){
    id = i;
```

```
    name = n;
    }
    //creating three arg constructor
    Student5(int i,String n,int a){
    id = i;
    name = n;
    age=a;
    }
    void display(){System.out.println(id+" "+name+" "+age);}

    public static void main(String args[]){
    Student5 s1 = new Student5(111,"Karan");
    Student5 s2 = new Student5(222,"Aryan",25);
    s1.display();
    s2.display();
   }
}
```

output:

```
111 Karan 0
222 Aryan 25
```

```
**********************************************************************************
5)
//Java program to initialize the values from one object to another object.
class Student6{
    int id;
    String name;
    //constructor to initialize integer and string
```

```
    Student6(int i,String n){
    id = i;
    name = n;
    }
    //constructor to initialize another object
    Student6(Student6 s){
    id = s.id;
    name =s.name;
    }
    void display(){System.out.println(id+" "+name);}

    public static void main(String args[]){
    Student6 s1 = new Student6(111,"Karan");
    Student6 s2 = new Student6(s1);
    s1.display();
    s2.display();
   }
}
```

output:

```
111 Karan
111 Karan
```

****

6)

```
class Student7{
    int id;
    String name;
    Student7(int i,String n){
```

```
    id = i;
    name = n;
    }
    Student7(){}
    void display(){System.out.println(id+" "+name);}

    public static void main(String args[]){
    Student7 s1 = new Student7(111,"Karan");
    Student7 s2 = new Student7();
    s2.id=s1.id;
    s2.name=s1.name;
    s1.display();
    s2.display();
   }
}
```

output:

```
111 Karan
111 Karan
```

****

1)

```
// Java Program to illustrate calling a
// no-argument constructor
import java.io.*;

class Geek
{
    int num;
```

```
    String name;

    // this would be invoked while an object
    // of that class is created.
    Geek()
    {
        System.out.println("Constructor called");
    }
}

class GFG
{
    public static void main (String[] args)
    {
        // this would invoke default constructor.
        Geek geek1 = new Geek();

        // Default constructor provides the default
        // values to the object like 0, null
        System.out.println(geek1.name);
        System.out.println(geek1.num);
    }
}
```

output:

```
Constructor called
null
0
```

****

2)

```
// Parameterized Constructor
// Importing required input output class
import java.io.*;

// Class 1
class Name {
    // data members of the class.
    String name;
    int id;

    // Constructor would initialize data members
    // With the values of passed arguments while
    // Object of that class created
    Name(String name, int id)
    {
        this.name = name;
        this.id = id;
    }
}

// Class 2
class Main1 {
    // main driver method
    public static void main(String[] args)
    {
        // This would invoke the parameterized constructor.
```

```
        Name geek1 = new Name("adam", 1);
        System.out.println("GeekName :" + geek1.name
                           + " and GeekId :" + geek1.id);
    }
}
```

output:

```
GeekName :adam and GeekId :1
```

****

3)

```
// Java Program to illustrate constructor overloading
// using same task (addition operation ) for different
// types of arguments.
import java.io.*;

class Word
{
    // constructor with one argument
    Word(String name)
    {
        System.out.println("Constructor with one " +
                       "argument - String : " + name);
    }

    // constructor with two arguments
    Word(String name, int age)
    {
```

```
        System.out.println("Constructor with two arguments : " +
                " String and Integer : " + name + " "+ age);

    }

    // Constructor with one argument but with different
    // type than previous..
    Word(long id)
    {
        System.out.println("Constructor with one argument : " +
                                            "Long : " + id);
    }
}

class Main2
{
    public static void main(String[] args)
    {
        // Creating the objects of the class named 'Geek'
        // by passing different arguments

        // Invoke the constructor with one argument of
        // type 'String'.
        Word geek2 = new Word("Shikhar");

        // Invoke the constructor with two arguments
        Word geek3 = new Word("Dharmesh", 26);
```

```
        // Invoke the constructor with one argument of
        // type 'Long'.
        Word geek4 = new Word(325614567);
    }
}
```

output:

```
Constructor with one argument - String : Shikhar
Constructor with two arguments :  String and Integer : Dharmesh 26
Constructor with one argument : Long : 325614567
**********************************************************************************
```