**Practical No.6**

**Name**: Anuj Rajendra Mane

**Div**: SY A

**Roll No**: 65 **Batch**: A3

**PRN** **No**: 2162701242054

**Title: Program on Constructor**

**Theory Of Constructor:**

Constructor is a special method of a class which is invoked automatically whenever an object is created. It has same name that of a class. A constructor initializes an object immediately upon creation. A constructor has no return type; not even void. It cannot be abstract, final, native ,static or synchronized. this keyword refers another constructor in same class. A super keyword will call constructor of super class constructors are of two type:

1.Default constructor

2.Parameterised constructor

**1. Default constructor:**

A constructor that accepts no parameters is called default constructor. If no constructor is defined for a class java system automatically generates the default constructor. A default constructor is called when an instance is created for a class. The default constructor automatically initializes all instance variables to zero.

**Example**:

Default constructor is to be set with values such as eye color, skin color and mouse color. Consider human as a class. The birth of a child is like initialising an object for that class.

There is no kid in this world without a name. So naming ceremony is the functionality inside a constructor, which will be done at the time of initialisation. Hence every kid gets a name, necessarily!. Constructor in java is used to create the instance of the class.

**Program:**

class demo {

int x, y;

float z;

demo() {

x = 1;

y = 2;

z = 3;

}

void display() {

System.out.println("Values of x, y and z are:" + x + " " + y + " " + z);

}

}

class Demomain {

public static void main(String args[]) {

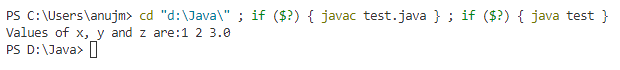
demo d1 = new demo(); // this is a call for the above default constructor

d1.display();

}

}

**Output**:

**]**

**2.Parameterized constructor:**

A constructor that takes arguments as parameters is called as parameterized constructor.

class demo {

int x;

int y;

float z;

demo(int x1, int y1, int z1) {

x = x1;

y = y1;

z = z1;

}

void display() {

System.out.println("Values of x, y and z are:" + x + " " + y + " " + z);

}

}

class Demomain1 {

public static void main(String args[]) {

demo d1 = new demo(1, 2, 3); // this is a call for the above parameterized constructor

d1.display();

}

}

**Output**:



**3. Java Copy Constructor:**

There is no copy constructor in java. But, we can copy the values of one object to another like copy constructor in C++. There are many ways to copy the values of one object into another in java. They are:

• By constructor

• By assigning the values of one object into another

• By clone() method of Object class

In this example, we are going to copy the values of one object into another using java constructor.

**Program:**

class Student6 {

int id;

String name;

Student6(int i, String n) {

id = i;

name = n;

}

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, "Anuj");

Student6 s2 = new Student6(s1);

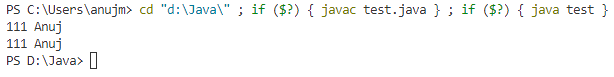
s1.display();

s2.display();

}

}

**Output**:



**Conclusion**:

I can learn the all concepts of constructor also learning the types of constructor and implementing all types of programs of constructor.

**Completion Date: Co-Ordinator Sign:**