**OOPs concepts in Java – 1**

**Practical 1**

1. Write a program to create a class and implement a default, overloaded and copy Constructor.

**Filename:** constructor\_example.java

**Code:**

**class** MyClass {

**private** **int** number;

**private** String text;

// Default constructor

**public** MyClass() {

number = 0;

text = "Default";

}

// Overloaded constructor

**public** MyClass(**int** num, String txt) {

number = num;

text = txt;

}

// Copy constructor

**public** MyClass(MyClass obj) {

number = obj.number;

text = obj.text;

}

**public** **void** display() {

System.***out***.println("Number: " + number + ", Text: " + text);

}

}

**public** **class** constructor\_example {

**public** **static** **void** main(String[] args) {

// Default constructor usage

MyClass obj1\_default = **new** MyClass();

System.***out***.print("Default Constructor: ");

obj1\_default.display();

// Overloaded constructor usage

MyClass obj2\_overloaded = **new** MyClass(10, "Overloaded");

System.***out***.print("Overloaded Constructor: ");

obj2\_overloaded.display();

// Copy constructor usage

MyClass obj3\_copy = **new** MyClass(obj2\_overloaded); // using copy constructor

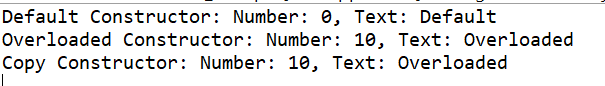
System.***out***.print("Copy Constructor: ");

obj3\_copy.display();

}

}

Output:



1. Write a program to create a class and implement the concepts of Method Overloading

**Filename:** TestMethodOverload.java

Code:

**public** **class** TestMethodOverload {

**public** **static** **void** main(String[] args) {

*add*(12,43);

*sub*(100.3f,23.3f);

*div*(12000l,2l);

}

**static** **int** add(**int** a,**int** b) {

**int** c=a+b;

System.***out***.println("Addition: "+c);

**return** c;

}

**static** **float** sub(**float** x,**float** y) {

**float** z=x-y;

System.***out***.println("Subtrat: "+z);

**return** z;

}

**static** **long** div(**long** p,**long** q) {

**long** r=p/q;

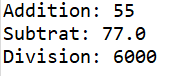
System.***out***.println("Division: "+r);

**return** r;

}

}

Output:



1. Write a program to create a class and implement the concepts of Static methods

Filename: StaticExample.java

Code:

**public** **class** StaticExample {

**private** **static** **int** *count* = 0; // static variable to count the number of instances

**public** StaticExample() {

*count*++; // Increment count every time a new instance is created

}

**public** **static** **int** getCount() {

**return** *count*; // Static method to get the count of instances

}

**public** **static** **void** main(String[] args) {

// Creating instances of StaticExample

StaticExample obj1 = **new** StaticExample();

StaticExample obj2 = **new** StaticExample();

StaticExample obj3 = **new** StaticExample();

// Calling static method getCount() to get the count of instances

System.***out***.println("Number of instances created: " + StaticExample.*getCount*());

}

}

Output:

Number of instances created: 3