**Aim:** a. Write a program to create a class and implement a default, overloaded and copy Constructor.

**Code:**

class MyClass

{

private int a;

public MyClass()

{

System.out.println("Default Constructor");

}

public MyClass(int value)

{

a=value;

System.out.println("Parameterized Constructor and value is:"+a);

}

public MyClass(MyClass other)

{

a = other.a;

System.out.println("Copy Constructor and value"+a);

}

}

public class pra1a

{

public static void main(String[] args)

{

MyClass obj1 = new MyClass();

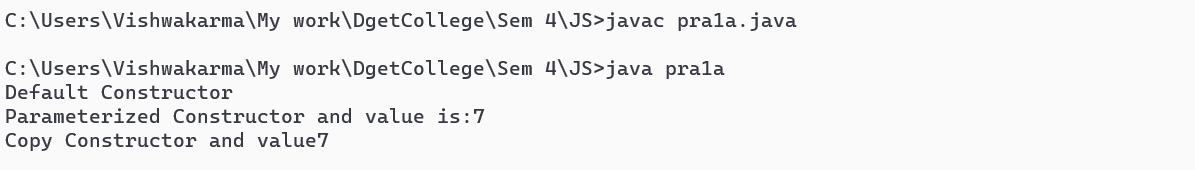
MyClass obj2 = new MyClass(7);

MyClass obj3 = new MyClass(obj2);

}

}

**Output:**



**Aim:** b. Write a program to create a class and implement the concepts of Method Overloading

**Code:**

class OperOver

{

public int add(int a,int b)

{

return a+b;

}

public int add(int a,int b,int c)

{

return a+b+c;

}

}

public class pra1b

{

public static void main(String[] args)

{

OperOver obj= new OperOver();

int sum1=obj.add(5,10);

int sum2=obj.add(5,10,15);

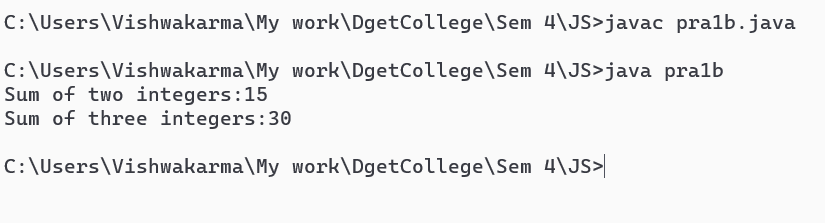
System.out.println("Sum of two integers:"+sum1);

System.out.println("Sum of three integers:"+sum2);

}

}

**Output:**



**Aim:** c. Write a program to create a class and implement the concepts of Static methods

**Code:**

class DemoStaticMethods

{

public static int add(int a,int b)

{

return a+b;

}

public static int sub(int a,int b)

{

return a-b;

}

}

public class pra1c

{

public static void main(String[] args)

{

int sum = DemoStaticMethods.add(8,4);

int diff = DemoStaticMethods.sub(7,6);

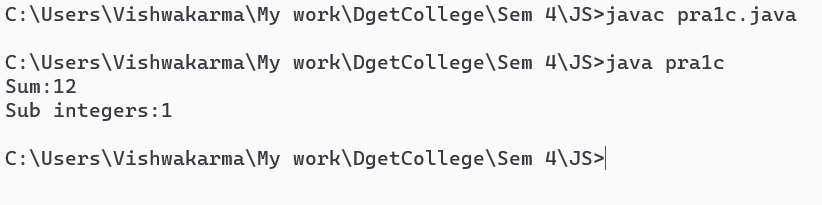
System.out.println("Sum:"+sum);

System.out.println("Sub integers:"+diff);

}

}

**Output:**

****