PRACTICAL ASSIGNMENT NO - I

1. Write a Java menu driven program using switch case to perform all Arithmetic operations on two Integers.

```
import java.util.*;

class Q1

{

   public static void main(String args[])

   {

      Scanner sc = new Scanner(System.in);

      int a,b,c,ch;

      do

      {


System.out.println("1:Addition\n2:Substraction\n3:Multiplication\n4:Division\n5:Modulation\n6:exit..");

          System.out.print("enter your choice=");

         ch=sc.nextInt();

         switch(ch)

         {

            case 1:

                  System.out.print("enter your 1st no=");

                  a=sc.nextInt();

                  System.out.print("enter your 2nd no=");

                  b=sc.nextInt();

                  c=a+b;

                  System.out.println("Addition="+c);
```

```
        break;
case 2:
        System.out.print("enter your 1st no=");
        a=sc.nextInt();
        System.out.print("enter your 2nd no=");
        b=sc.nextInt();
        c=a-b;
        System.out.println("Substraction="+c);
        break;
case 3:
        System.out.print("enter your 1st no=");
        a=sc.nextInt();
        System.out.print("enter your 2nd no=");
        b=sc.nextInt();
        c=a*b;
        System.out.println("Multiplication="+c);
        break;
case 4:
        System.out.print("enter your 1st no=");
        a=sc.nextInt();
        System.out.print("enter your 2nd no=");
        b=sc.nextInt();
        c=a/b;
        System.out.println("Division="+c);
        break;
case 5:
        System.out.print("enter your 1st no=");
```

```
                a=sc.nextInt();
                System.out.print("enter your 2nd no=");
                b=sc.nextInt();
                c=a%b;
                System.out.println("Modulation="+c);
                break;
            case 6:
                break;
        }
    }while(ch!=6);
  }
}
```

Output:

```
1:Addition
2:Substraction
3:Multiplication
4:Division
5:Modulation
6:exit..
enter your choice=1
enter your 1st no=10
enter your 2nd no=10
Addition=20
```

2.Write a Java program to demonstrate local, instance and static variables.

```
class Q2
{
   int instanceVar=10;
   static int staticVar=20;
   void display()
   {
      int localVar=30;

      System.out.println("Instance Variable="+instanceVar);
      System.out.println("Static Variable="+staticVar);
      System.out.println("Local Variable="+localVar);
   }
   public static void main(String args[])
   {
      Q2 obj=new Q2();
      obj.display();
   }
}
```

Output:

Instance Variable=10

Static Variable=20

Local Variable=30

3. Write a program to print odd numbers between 50 to 100 in reverse order

```
class Q3
{
   public static void main(String args[])
   {
      int i;
      for(i=100;i>=50;i--)
      {
         if(i%2!=0)
         {
            System.out.print("\t"+i);
         }
      }
   }
}
```

Output:

```
    99    97    95    93    91    89    87    85    83    81    79    77    75    73    71    69 67
65    63    61    59    57    55    53    51
```

4. Write a Java program which prints given number is prime or not.

```
import java.util.*;
class Q4
{
   public static void main(String args[])
   {
      Scanner sc = new Scanner(System.in);
      int n,s=0,i;
      System.out.print("enter any no=");
      n=sc.nextInt();
      for(i=1;i<=n;i++)
      {
         if(n%i==0)
         {
            s++;
         }
      }
      if(s==2)
      {
         System.out.println("number is prime");
      }
      else
      {
         System.out.println("number is not prime");
```

```
    }


  }
}
```

Output:

enter any no=11

number is prime

5. Write a Java program which accepts consumed units of electricity And Calculate the bill according to following table.

Unit range	Rate in Rs.
1 to 100	10 per unit
101 to 200	15 per unit
201 to 300	20 per unit
Above 300	25 per unit

```
import java.util.*;
class Q5
{
   public static void main(String args[])
   {
```

```
        Scanner sc = new Scanner(System.in);
        int unit;
        System.out.print("enter your units of electricity bill=");
        unit=sc.nextInt();
        if(unit>1 && unit<=100)
        {
           unit=unit*10;
        }
        else if(unit>100 && unit<=200)
        {
           unit=unit*15;
        }
        else if(unit>200 && unit<=300)
        {
           unit=unit*20;
        }
        else if(unit>300)
        {
           unit=unit*25;
        }
        System.out.print("your electricity bill="+unit);
    }
}
```

Output:

enter your units of electricity bill=12

your electricity bill=120

6 Write a Java program to which input elements from user for 1D array and display it.

```
import java.util.*;
class Q6
{
   public static void main(String args[])
   {
      Scanner sc=new Scanner(System.in);
      int x[]=new int[3];
      int i;
         System.out.println("enter array elements");
         for(i=0;i<3;i++)
         {
            x[i]=sc.nextInt();
         }
         System.out.println(" array elements are-:");
         for(i=0;i<3;i++)
         {
            System.out.print("\t"+x[i]);
         }
   }
}
```

Output:

enter array elements

1

2

3

array elements are-:

1 2 3

7. Write a Java program to implement jagged array.

```
class Q7
{
   public static void main(String[] args)
   {
      int z[ ][ ]=new int [2][ ];
      z[0]=new int[4];
      z[1]=new int[2];
      z[0][0]=10;
      z[0][1]=11;
      z[0][2]=12;
      z[0][3]=13;
      for(int i=0;i<=3;i++)
```

```
        {
            System.out.print("\t"+z[0][i]);
        }
        System.out.println();
        System.out.println();
        z[1][0]=5;
        z[1][1]=6;
        for(int i=0;i<=1;i++)
        {
            System.out.print("\t"+z[1][i]);
        }
    }
}
```

Output:

```
 10     11     12     13

     5     6
```

8. Write a Java program to implement multidimensional array.

```
import java.util.*;
class A
{
    public static void main(String args[])
```

```
  {
     Scanner sc=new Scanner(System.in);
     int x[][]=new int[2][2];
     int i,j;
        System.out.println("enter array elements");
        for(i=0;i<3;i++)
        {
           for(j=0;j<3;j++)
           {
              x[i][j]=sc.nextInt();
           }
        }
        System.out.println(" array elements are-:");
        for(i=0;i<3;i++)
        {
           for(j=0;j<3;j++)
           {
              System.out.print("\t"+x[i][j]);
           }
           System.out.println("");
        }
  }
}
```

Output:

1

```
2
3
4
5
6
7
8
9
9
10
11
 array elements are-:
      1     2     3     4     5     6
      7     8     9     9     10    11
```

9. Write a Java program to find length of array.

```
class Q9
{
   public static void main(String s[])
   {
      int a[ ]={7,3,9,7,2,4};
      int b[ ][ ]=new int[5][7];
      int c[ ][ ][ ]=new int[9][3][2];
      int x,y,z;
```

```
        x=a.length;
        y=b.length;
        z=c.length;
        System.out.println("1D array size="+x);
        System.out.println("2D array size="+y);
        System.out.println("Multi array size="+z);
    }
}
```

Output:

1D array size=6

2D array size=5

Multi array size=9

10. Write a java class to accept student details like roll no., name and address and display it.

```
import java.util.*;
class Q10
{
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        int rollno;
```

```
        String name;
        String address;
        System.out.println("enter your rollno=");
        rollno=sc.nextInt();
        System.out.println("enter your name=");
        name=sc.next();
        System.out.println("enter your address=");
        address=sc.next();
        System.out.println("enter your rollno: "+rollno);
        System.out.println("enter your name: "+name);
        System.out.println("enter your address: "+address);
    }
}
```

Output:

```
enter your rollno=
11
enter your name=
mahii
enter your address=
mm
enter your rollno: 11
enter your name: mahii
enter your address: mm
```

11. Write a Java program which accepts emp id and emp name, use them as actual parameter and send them to the constructor, after that display the record from constructor.

```
import java.util.*;


class Employee
{
   int empId;
   String empName;
   Employee(int id, String name)
   {
      this.empId = id;
      this.empName = name;
      display();
   }
   void display() {
      System.out.println("Employee ID: " + empId);
      System.out.println("Employee Name: " + empName);
   }
}


public class Q11
{
   public static void main(String[] args)
   {
```

```
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Employee ID: ");
        int id = sc.nextInt();
        System.out.println("");
        System.out.print("Enter Employee Name: ");
        String name = sc.next();
        new Employee(id, name);
    }
}
```

Output:

Enter Employee ID: 1

Enter Employee Name: mahii

Employee ID: 1

Employee Name: mahii

12. Write a Java program of constructor overloading.

```
class person
{
   int age;
   String name,add;
```

```
    double sal;
    person( )
    {
       age=70;
       name= "Sachin";
       add= "Pune";
       sal= 4568.50;
    }
    person(int a, String n )
    {
       age=a;
       name=n;
       add= "MUMBAI";
       sal= 54508.75;
    }
    void show( )
    {
    System.out.println("Age="+age);
    System.out.println("Name="+name);
    System.out.println("Address="+add);
    System.out.println("Salary="+sal);
    }
}

class Q12
{
    public static void main(String args[])
```

```
    {
        person p=new person( );
        person q=new person(45,"mahii");
        p.show( );
        q.show( );
    }
}
```

Output:

```
Age=70
Name=Sachin
Address=Pune
Salary=4568.5
Age=45
Name=mahii
Address=MUMBAI
Salary=54508.75
```

13. Write a Java program which refers current executing object.

a)inside the constructor. b) inside the member function.

```
class Sample
{
    int value;
```

```
    Sample(int value)
    {
        this.value = value;
        System.out.println("Constructor executed for: " + this);
    }

    void display()
    {
        System.out.println("Value: " + value);
        System.out.println("Method executed for: " + this);
    }
}

public class Q13
{
    public static void main(String[] args)
    {
        Sample obj = new Sample(50);
        obj.display();
    }
}
```

Output:

Constructor executed for: Sample@1db9742

Value: 50

Method executed for: Sample@1db9742

14.Write a Java program to implement static method and static block.

```
class StaticExample
{
   static int staticValue;

   static
   {
      staticValue = 100;
      System.out.println("Static block value: " + staticValue);
   }
   static void display()
   {
      System.out.println("Static method value: " + staticValue);
   }
}

public class Q14
{
   public static void main(String[] args)
   {
      StaticExample.display();
   }
```

```
}
```

Output:

Static block value: 100

Static method value: 100

15.Write a java program to demonstrate user defined package.

```
package myPackage;


class MyClass
{
   void display()
   {
      System.out.println("Inside User Defined Package");
   }
}


class Q15
{
   public static void main(String[] args)
   {
      myPackage.MyClass obj = new myPackage.MyClass();
      obj.display();
   }
```

```
}
```

Output:

Inside User Defined Package

16.Write a java program to implements all types of inheritance.

```
class Parent
{
   void showParent()
   {
      System.out.println("This is Parent class");
   }
}

class Child extends Parent
{
   void showChild()
   {
      System.out.println("This is Child class (Single Inheritance)");
   }
}

class GrandChild extends Child
```

```
{
   void showGrandChild()
   {
      System.out.println("This is GrandChild class (Multilevel Inheritance)");
   }
}

class Sibling extends Parent
{
   void showSibling()
   {
      System.out.println("This is Sibling class (Hierarchical Inheritance)");
   }
}

interface FirstInterface
{
   void showFirst();
}

interface SecondInterface
{
   void showSecond();
}

class MultipleInheritanceDemo implements FirstInterface, SecondInterface
{
```

```
    public void showFirst()
    {
        System.out.println("This is First Interface");
    }

    public void showSecond()
    {
        System.out.println("This is Second Interface");
    }
}

class HybridChild extends Child implements FirstInterface
{
    public void showFirst()
    {
        System.out.println("This is HybridChild implementing FirstInterface");
    }
}

public class Q16
{
    public static void main(String[] args)
    {
        GrandChild grandChild = new GrandChild();
        grandChild.showParent();
        grandChild.showChild();
        grandChild.showGrandChild();
```

```
        Sibling sibling = new Sibling();
        sibling.showParent();
        sibling.showSibling();

        MultipleInheritanceDemo multiDemo = new MultipleInheritanceDemo();
        multiDemo.showFirst();
        multiDemo.showSecond();

        HybridChild hybrid = new HybridChild();
        hybrid.showParent();
        hybrid.showChild();
        hybrid.showFirst();
    }
}
```

Output:

```
This is Parent class
This is Child class (Single Inheritance)
This is GrandChild class (Multilevel Inheritance)
This is Parent class
This is Sibling class (Hierarchical Inheritance)
This is First Interface
This is Second Interface
This is Parent class
```

This is Child class (Single Inheritance)

This is HybridChild implementing FirstInterface

17.Write a java program to stop inheritance.

```
final class FinalClass
{
   void display()
   {
      System.out.println("This is a final class, inheritance is not allowed.");
   }
}

public class Q17
{
   public static void main(String[] args)
   {
      FinalClass obj = new FinalClass();
      obj.display();
   }
}
```

Output:

This is a final class, inheritance is not allowed.

18.Write a java program to override method.

```
class Parent
{
   void show()
   {
      System.out.println("This is the Parent class method.");
   }
}

class Child extends Parent
{
   @Override
   void show()
   {
      System.out.println("This is the Child class overriding the Parent method.");
   }
}

public class Q18
{
   public static void main(String[] args)
   {
      Parent obj1 = new Parent();
      obj1.show();
```

```
        Child obj2 = new Child();
        obj2.show();

        Parent obj3 = new Child();
        obj3.show();
    }
}
```

Output:

This is the Parent class method.

This is the Child class overriding the Parent method.

This is the Child class overriding the Parent method.

19.Write a java program to implement abstract class and abstract method.

```
abstract class Myabs
{
    abstract void calculate(double x);
}

class sub1 extends Myabs
{
    void calculate(double a)
```

```
    {
        System.out.println("Square="+(a*a));
    }
}

class sub2 extends Myabs
{
    void calculate(double a)
    {
        System.out.println("SquareRoot="+Math.sqrt(a));
    }
}

class sub3 extends Myabs
{
    void calculate(double a)
    {
        System.out.println("Cube="+(a*a*a));
    }
}

class Q19
{
    public static void main(String as[])
    {
        sub1 t1=new sub1();
        sub2 t2=new sub2();
```

```
        sub3 t3=new sub3();
        t1.calculate(5);
        t2.calculate(36);
        t3.calculate(4);
    }
}
```

Output:

Square=25.0

SquareRoot=6.0

Cube=64.0

20.Write a java program to implement any interface.

```
interface A
{
    int p=10;
    void calculate();
}

interface B
{
    int p=20;
    void calculate();
```

```
}
interface C
{
   int p=30;
   void calculate();
}

class D implements A,B,C
{
   public void calculate()
   {
      int z;
      z=A.p+B.p+C.p;
      System.out.println("Addition="+z);
   }
}

class Q20
{
   public static void main(String s[])
   {
      D obj=new D();
      obj.calculate();
   }
}
```

Output:

Addition=60

21.Write a java program to implement wrapper class

```
class Q21
{
   public static void main(String[] args)
   {
      Integer myInt = 5;
      Double myDouble = 5.99;
      Character myChar = 'A';
      System.out.println(myInt);
      System.out.println(myDouble);
      System.out.println(myChar);
   }
}
```

Output:

5

5.99

A