Java Assignment-2

1.Write A Program To Implement Super as Variable,Constructor and Method.

```
class sup {
   int x, y;

   sup(int x, int y) {
      this.x = x;
      this.y = y;
   }

   void display() {
      System.out.println("Super Class Method is called");
   }
}

class sub extends sup {
   sub(int a, int b) {
      super(a, b);
   }

   public void dis() {
      super.display();
      System.out.println("Sub Class Method is Called: " + super.x + " " + super.y);
   }
}

public class assignment2_1 {
   public static void main(String[] args) {
      sub obj = new sub(10, 20);
      obj.dis();
   }
}
```

Ouput:-
Super Class Method is called
Sub Class Method is Called: 10 20

2.Write A program To Implement Super As A Constructor

```
class sup {
   int x, y;
```

```
    sup(int x, int y) {

        this.x = x;
        this.y = y;
        System.out.println("Super Class Constructor is called: "+this.x+" "+this.y);
    }
}

class sub extends sup {
    sub(int a, int b) {
        super(a, b);
    }
}
public class assignment2_2 {
    public static void main(String[] args) {
        sub obj=new sub(10, 20);
    }
}
```

Ouput:-
Super Class Constructor is called: 10 20

3.Write A Program To Implement Super As A Method

```
class sup {
    void display() {
        System.out.println("Super Class Method is Called");
    }
}

class sub extends sup {
    public void dis() {
        super.display();
        System.out.println("Sub Class Method is Called: ");
    }
}

public class assignment2_3 {
    public static void main(String[] args) {
        sub obj=new sub();
        obj.dis();
    }
```

```
}
```

Ouput:-
Super Class Method is Called
Sub Class Method is Called:

4.Write A Program To Implement Final As A Variable

```
public class assignment2_4 {
    public static void main(String[] args) {
        final int a = 10;
        int b = 20, c;
        // a = 50; //This Will Show Error Because Final Varibale's Value Can't Change
        c = a + b;
        System.out.println("The Result Is : " + c);
    }
}
```

Output:-
The Result Is : 30

5.Write A Program To Implement Final As A Method

```
class parent {
    final void display() {
        System.out.println("Parent Class ");
    }
}

class child extends parent {
    // void display() {
    // System.out.println("Child Class");
    // } // This Method Can't be Override Because In Parent Class This Method is
    // Final method.

    void dis() {
        System.out.println("Child Class");
    }
}

public class assignment2_5 {
    public static void main(String[] args) {
```

```
        child obj = new child();
        obj.display();
        obj.dis();
    }
}
```

Ouput:-
Parent Class
Child Class

6.Write A Program To Implement Final As A Class

```
final class a1 {
    void sum() {
        System.out.println("The Sum Is " + (10 + 20));
    }
}

// class a2 extends a1{
// }// The a1 class can't be extended because a1 class is final class
class a2 {
    void display() {
        System.out.println("The Sum Is: " + (50 + 50));
    }
}

public class assignment2_6 {
    public static void main(String[] args) {
        a1 obj1 = new a1();
        a2 obj2 = new a2();
        obj1.sum();
        obj2.display();
    }
}
```

Output:-
The Sum Is 30
The Sum Is: 100

7.Write A Program To Implement Method Overloading.

```
class meth {
    void sum(int a) {
```

```
        System.out.println("The Sum Is " + (10 + a));
    }

    void sum(int a, int b) {
        System.out.println("The Sum Is: " + (10 + a + b));
    }

    void sum(int a, int b, int c) {
        System.out.println("The Sum Is: " + (10 + a + b + c));
    }
}


public class assignment2_7 {
    public static void main(String[] args) {
        meth obj = new meth();
        obj.sum(20);
        obj.sum(20, 20);
    }
}
```

Output:-
The Sum Is 30
The Sum Is: 50

8.Write A Java Program To Implement Method Overriding

```
class parent{
    void sum(int a,int b){
        int c=a+b;
        System.out.println("The Parent Method Call And The Sum Is: "+c);
    }
}

class child extends parent{
    void sum(int a,int b){
        int c=a+b;
        System.out.println("The Child Method Call And The Sum Is: "+c);
    }
}
public class assignment2_8 {
    public static void main(String[] args) {
        child obj=new child();
```

```
        obj.sum(15, 25);
    }
}
```

Output:-
The Child Method Call And The Sum Is: 40

9.Write A Java Program To Demonstrate Static As A Variable

```
class stst{
    static int a=10;
    void display()
    {
        a++;

        System.out.println("The Value Is: "+a);
    }
}
public class assignment2_9 {
    public static void main(String[] args) {
        stst obj1=new stst();
        stst obj2=new stst();
        stst obj3=new stst();
        obj1.display();
        obj2.display();
        obj3.display();
    }
}
```

Output:-
The Value Is: 11
The Value Is: 12
The Value Is: 13

10.Write A Java Program To Demonstrate Static As A Method

```
public class assignment2_10 {
    static void display(){
        System.out.println("Hello My Name Is Programmer");
    }
```

```
    public static void main(String[] args) {
        assignment2_10.display();
    }
} public class assignment2_10 {
    static void display(){
        System.out.println("Hello My Name Is Programmer");
    }
    public static void main(String[] args) {
        assignment2_10.display();
    }
}
```

Output:-
Hello My Name Is Programmer

11.Write A Java Program To Demonstrate Static As A Block

```
class demo{
    static int a=10,b=20,c;

    static{
        c=a+b;
        System.out.println("This Is Static Block And Value Is: "+c);
    }
}
public class assignment2_11 {
    public static void main(String[] args) {
        demo obj=new demo();
    }
}
```

Output:-
This Is Static Block And Value Is: 30

12.Write A Java Program To Implement Abstract Class

```
abstract class a1{
    void display(){
        System.out.println("Hello This Is Not Abstract Method");
    }
    abstract void dis();
}
class a2 extends a1{
```

```
    void dis(){
        System.out.println("Hello This Is Abstract Method");
    }
}
public class assignment2_12 {
    public static void main(String[] args) {
        a1 obj=new a2();
        obj.dis();
        obj.display();
    }
}
```

Output:-
Hello This Is Abstract Method
Hello This Is Not Abstract Method

13.Write A Java Program To Achieve Multiple Inheritance

```
interface a1{
    void method1();
}
interface a2{
    void method2();

}
class a3 implements a1,a2{
    public void method1(){
        System.out.println("Hello My Name Is Method1");
    }
    public void method2(){
        System.out.println("Hello My Name Is Method2");
    }
}
public class assignment2_13 {
    public static void main(String[] args) {
        a3 obj=new a3();
        obj.method1();
        obj.method2();
    }
}
```

Output:-
Hello My Name Is Method1

Hello My Name Is Method2

14.Write A Java Program To Achieve Fully Abstraction.

```
interface full{
   void method1();
   void method2();
}
class a1 implements full{
   public void method1(){
      System.out.println("Hello My Name Is Method1");
   }
   public void method2(){
      System.out.println("Hello My Name Is Method2");
   }
}
public class assignment2_14 {
   public static void main(String[] args) {
      a1 obj =new a1();
      obj.method1();
      obj.method2();
   }
}
```

Output:-
Hello My Name Is Method1
Hello My Name Is Method2

15.Write A Java Program To Achieve Check Exception

```
import java.io.DataInputStream;
import java.io.IOException;

public class assignment2_15 {
   public static void main(String[] args) throws IOException {
      DataInputStream dt=new DataInputStream(System.in);
      try {
         System.out.print("Enter Value: ");
         String inputLine = dt.readLine();
         int ab = Integer.parseInt(inputLine);
         System.out.println("The Value Is: " + ab);
      } catch (NumberFormatException e) {
         System.err.println("Invalid input. Please enter a valid integer.");
```

```
        }
    }
}
```

Output:-
Note: assignment2_15.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
Enter Value: 10
The Value Is: 10

16.Write A Java Program To Achieve Unchecked Exception

```
public class assignment2_16 {
    public static void main(String[] args) {
        int a=20;
        a=a/0;
        System.out.println("The Value is : "+a);
    }
}
```

Output:-
Exception in thread "main" java.lang.ArithmeticException: / by zero
at assignment2_16.main(assignment2_16.java:4)

17.Write A Java Program To Implement Finally Block

```
import java.util.Scanner;

public class assignment2_17 {

    public static void main(String[] args) {
        int a;
        try {
            Scanner scan = new Scanner(System.in);
            System.out.print("Enter Value: ");
            a = scan.nextInt();
            a = a / 0;
            System.out.println("The Value Is: " + a);
        } catch (ArithmeticException e) {
            System.out.println("Zero Can't Be Divided");
        } finally {
            System.out.println("Program Was Done");
        }
```

```
    }
}
```

Output:-
Enter Value: 10
Zero Can't Be Divided
Program Was Done

18.Write A Java Program To Achieve User Defined Exception

```
import java.util.Scanner;

class caught extends ArithmeticException{
    @Override
    public String toString() {
      return "Zero Can't Be Divided";
    }
}
public class assignment2_18 {
    public static void main(String[] args) {
        int a,n;
        Scanner scan = new Scanner(System.in);
        System.out.print("Enter First Value: ");
        a = scan.nextInt();
        System.out.print("Enter Second Value: ");
        n = scan.nextInt();
        if (n==0) {
            try {
                throw new caught();
            } catch (ArithmeticException e) {
                System.out.println(e.toString());
                return;

            } finally {
                System.out.println("Program Was Done");
            }
        }
        else{
            a=a/n;
            System.out.println("The Value Is: "+a);
        }
    }
```

```
}
```

Output:-
Enter First Value: 10
Enter Second Value: 0
Zero Can't Be Divided
Program Was Done