```java
/*1.Write a Java Program to implement Super as Variable, Constructor
and Method. */

class parent {
    int i = 10;

    parent() {
        System.out.println("Constructor of Parent.");
    }

    void meth() {
        System.out.println("Method Of Parent Class.");
    }
}

class child extends parent {
    int i = 50;

    child() {
        super();
        System.out.println("Child Class Constructor.");
    }

    void display() {
        super.meth();
        System.out.println("Value of Parent Variable :" + super.i);
    }

}

public class j21 {
    public static void main(String[] args) {

        child c1 = new child();
        c1.display();
    }
}
```

OUTPUT :-

```
Constructor of Parent.
Child Class Constructor.
Method Of Parent Class.
Value of Parent Variable :10
```

```java
/*2.    Write a Java Program to implement Super as Constructor. */
class parent {
    parent() {
        System.out.println("Constructor Of parent class.");
    }
}

class child extends parent {
    child() {
        System.out.println("Constructor Of child class.");
    }
}

public class j22 {
    public static void main(String[] args) {
        child c1 = new child();

    }

}
```

OUTPUT:-

```
Constructor Of parent class.
Constructor Of child class.
```

```java
/*3.Write a Java Program to implement Super as Method. */
class parent {
    void fun() {
        for (int i = 1; i <= 10; i++) {
            System.out.println(i);
        }
    }
}

class child extends parent {
    void fun() {
        System.out.println("child class method.");
        super.fun();

    }
}

public class j23 {
    public static void main(String[] args) {
        child c1 = new child();
        c1.fun();
    }

}
```

OUTPUT:-

```
child class method.
1
2
3
4
5
6
7
8
9
10
```

```java
/*4.Write a Java Program to Implement Final as Variable. */
class parent {
    final int i = 10;

    void fun() {
        // i = 20;
        System.out.println("I :" + i);
    }
}

public class j24 {

    public static void main(String[] args) {
        parent p = new parent();
        p.fun();
    }
}
```

OUTPUT:-

```
I :10
```

```java
/*
 * 5.Write a Java Program to Implement Final as Method.
 */

class parent {
    final void disp() {
        System.out.println("This Is Final Method");
    }
}

public class j25 {

    public static void main(String[] args) {

        parent p1 = new parent();
        p1.disp();
    }
}
```

OUTPUT:-

```
This Is Final Method
```

```java
/**
 * 6. Write a Java Program to Implement Final as Class.
 */
final class parent {
    void disp() {
        System.out.println("This Is Final Class");
    }
}

public class j26 {
    public static void main(String[] args) {
        parent p1 = new parent();
        p1.disp();
    }

}
```

OUTPUT:-

```
This Is Final Class
```

```java
/*

  7. Write a Java Program to Implement Method Overloading.
 */
class parent {
    void add(int a, int b) {
        int sum = a + b;
        System.out.println("sum = " + sum);
    }

    void add(int a, int b, int c) {
        int sum = a + b + c;
        System.out.println("sum = " + sum);
    }

    void add(double a, double b) {
        double sum = a + b;
        System.out.println("sum = " + sum);
    }

    void add(String a, String b) {
        String sum = a + b;
        System.out.println("sum = " + sum);
    }
}

public class j27 {
    public static void main(String[] args) {
        parent p1 = new parent();
        p1.add(12, 25);
        p1.add(10, 20, 30);
        p1.add(1.5, 2.5);
        p1.add("Jignesh", "Gupta");
    }

}
```

OUTPUT:-

```
sum = 37
sum = 60
sum = 4.0
sum = JigneshGupta
```

```java
/**
 * 8. Write a Java Program to Implement Method Overriding.
 */
class animal {
    void sound() {
        System.out.println("Animal Voice .");
    }
}

class dog extends animal {

    void sound() {
        System.out.println("Dog Bark");
    }
}

class cat extends animal {
    void sound() {
        System.out.println("Cat meow");
    }
}

public class j28 {
    public static void main(String[] args) {
        animal a1 = new animal();
        animal a2 = new cat();
        a1.sound();
        a2.sound();
    }
}
```

OUTPUT:-

```
Animal Voice .
Cat meow
```

```java
/**
 * 9.Write a Java Program to demonstrate Static as Variable
 */
class parent {
    String name;
    String Clgname = "VTP BCA COLLEGE";
    int rno;

    void getrecord(String n, int r) {
        name = n;
        rno = r;
    }

    void disp() {
        System.out.println("Name :" + name);
        System.out.println("Roll.no :" + rno);
        System.out.println("College : " + Clgname);
    }
}

public class j29 {
    public static void main(String[] args) {
        parent p1 = new parent();
        p1.getrecord("Rohit", 1);
        p1.disp();
        p1.getrecord("Virat", 2);
        p1.disp();
        p1.getrecord("Raina", 3);
        p1.disp();
        p1.getrecord("Manish", 4);
        p1.disp();
    }

}
```

OUTPUT:-

```
Name :Rohit
Roll.no :1
College : VTP BCA COLLEGE
Name :Virat
Roll.no :2
College : VTP BCA COLLEGE
Name :Raina
Roll.no :3
College : VTP BCA COLLEGE
Name :Manish
Roll.no :4
College : VTP BCA COLLEGE
```

```java
/**
 * 10. Write a Java Program to demonstrate Static as Method
 */
class parent {
    void disp() {
        System.out.println("This is Static Method.");
    }
}

public class j210 {
    public static void main(String[] args) {
        parent p1 = new parent();
        p1.disp();
    }

}
```

OUTPUT:-

```
This is Static Method.
```

```java
/**
 * 11.Write a Java Program to demonstrate Static as Block.
 */

class parent {
    static {
        System.out.println("This Is a Static Method");
    }
}

public class j211 {
    public static void main(String[] args) {

        parent p1 = new parent();

    }

}
```

OUTPUT:-

```
This Is a Static Method
```

```java
/**
 * 12. Write a Java Program to Implement Abstract Class.
 */

abstract class animal {
    abstract void sound();

    void sleep() {
        System.out.println("Slipping..");
    }

}

class dog extends animal {
    void sound() {
        System.out.println("Dog Barking");
    }
}

public class j212 {
    public static void main(String[] args) {
        dog d1 = new dog();
        d1.sound();
        d1.sleep();

    }

}
```

OUTPUT:-

```
Dog Barking
Slipping..
```

```java
/**
 * 13. Write a Java Program to Achieve Multiple Inheritance.
 */
interface parent1 {
    default void hair() {
        System.out.println("Parent1 Hair.");
    }
}

interface parent2 {
    default void face() {
        System.out.println("Parent2 Face");
    }
}

class child implements parent1, parent2 {
    void jack() {
        parent1.super.hair();
        parent2.super.face();
        System.out.println("Child Voice");

    }

}

public class j213 {
    public static void main(String[] args) {
        child c1 = new child();
        c1.jack();
    }

}
```

OUTPUT:-

```
Parent1 Hair.
Parent2 Face
Child Voice
```

```java
/**

 * 14.Write a Java Program to Achieve Fully Abstraction.

 */
abstract class ads {
    abstract void fun();
}

class j214 extends ads {
    void fun() {

        System.out.println("It Fully Abstraction.");
    }

    public static void main(String[] args) {
        j214 oj = new j214();

        oj.fun();

    }

}
```

OUTPUT:-

```
It Fully Abstraction.
```

```java
/**
 * 16.Write a Java Program to Achieve Uncheck Exception.
 */
public class j216 {
    public static void main(String[] args) {

        int i = 10, a = 0;

        try {
            int r = i / a;
            System.out.println("division : " + r);
        } catch (Exception e) {
            System.out.println(e);
        }
    }

}
```

OUTPUT:-

```
java.lang.ArithmeticException: / by zero
```

```java
/**
 * 17.Write a Java Program to Implement Finally Block.
 */
public class j217 {
    public static void main(String[] args) {

        int i = 10, a = 0;

        try {
            int r = i / a;
            System.out.println("division : " + r);
        } catch (Exception e) {
            System.out.println(e);
        } finally {
            System.out.println("Finally block Excecuted ");
        }
    }

}
```

OUTPUT:-

```
java.lang.ArithmeticException: / by zero
Finally block Excecuted
```

```java
/**
 * 18.Write a Java Program to Achieve User Define Exception.
 */
class ageexception extends Exception {

    ageexception(String message) {
        super(message);
    }
}

class inexception {
    static void excep(int age) throws ageexception {
        if (age < 0) {
            throw new ageexception("Age can not be negative.");
        }
    }
}

public class j218 {
    public static void main(String[] args) {
        try {
            inexception.excep(-2);
        } catch (ageexception e) {
            System.out.println("Error: " + e.getMessage());
        }
    }

}
```

OUTPUT:-

```
Error: Age can not be negative.
```