ASSIGNMENT-2

JAVA PROGRAMMING.

B. Abhilash
192365076
B.E.L.S.E.

Assignment 2 (Inheritance)

Single inheritance.

```
class A {
    public void disp(){
        System.out.println("parent class A");
    }
}
class B extends A {
    public void disp(){ child
        System.out.println("parent class B");
    }
}
public class main {
    public static void main(String[] args){
    b obj = new b();
    obj.disp1();
    obj disp();
    }
}
```

Multilevel inheritance

```
class a {
    public void disp(){
        System.out.println("Parent class
                            A");
    }
}
```

```
}
class a {
    public void disp(){
        System.out.println("Child class B");
    }
}

class b extends a{
    public void disp1(){
        System.out.println("Child class B");
    }
}

public class main{
    public static void main(String[] args){
        C obj = new();
        obj.disp();
        obj.disp();
        obj.disp2();
    }
}
```

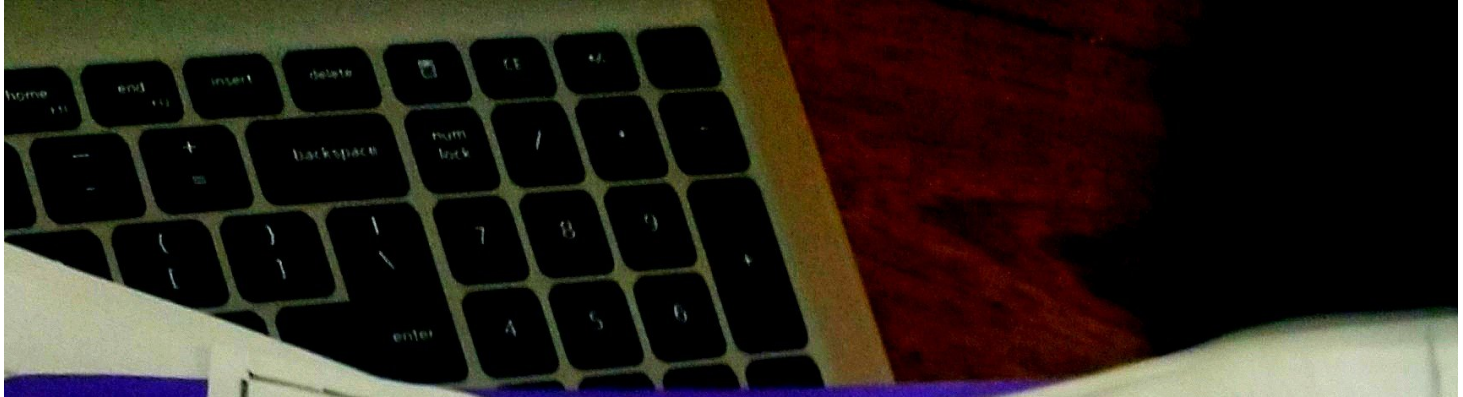
Hierarchical Inheritance.

```
class a {
    a(){
    System.out.println("parent class A");
    }
```

```
class b extends a {
    b(){
        System.out.println("Child class B");
    }
}

class c extends a {
    c(){
        System.out.println("Child class c");
    }
}

public class main{
    public static void main(String[] args){
        cobj = new c();
        bobj = new b();
    }
}
```

Hybrid Inheritance

```
class a{
    public void dispA(){
    System.out.println("class A");
    }
}

class b extends a{
    public void dispB(){
        System.out.println("class B");
    }
}
```



```
class C extends B {
    public void dispC() {
        System.out.println("class C");
    }
}
class D extends B {
    public void dispD() {
        System.out.println("class D");
    }
}
class main {
    public static void main(String[] args) {
        C obj = new C();
        obj = dispA();
        obj : dispB()
        D obj = new D();
        obj.dispA();
        obj.dispB();
        obj dispD();
    }
}
```

class A
class B
class C
class A
class B
class D

Multiple inheritance

```
interface A{
    int a=10;
    void dispA();
}
interface B.
    int b= 5;
    void dispB();
        System.out.println("a = "+a);
public void dispB(){
        System.out.println("b = "+b);
public void dispC(){
    System.out.println("c = "+c);
    }
    void process(){
        System.out.println("a+b+c"+a+b+c);
        }
    }
class main{
    public static void main
                    (String[] args){
    C obj = new C();
        obj.dispA();
        obj.dispB();
```

```
obj.disp();
obj.disp();
}
}
```

O/P:- a=10
b=5
c=15

a+b+c=30

Exception handling.

Arithematic Exception.

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

        int a=5, b=0;
        try {
            System.out.println(a/b);
            }
        catch(Arithematic Expression e){
            System.out.println("Divide by 0");
                }
            }
        }
```

Array index out of bound. exception.

```
public class main {
    public static Void main(String[] args) {
        try {
            int a[] = {1,2,3},
            System.out.println(a[10]);}
```

```
catch(ArrayIndexoutofBoundsException e){
    System.out.println("Array Index out
        of bounds Exception: "+
                    e.getmessage
                    ());
}}}
```

O/P:- Arthematic exception
This is the finally block.

finally

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

try {
    int a = 50/0;
    System.out.println("There is no Error in
                        the try block");

}
catch (Arithmetic Exception e){
    System.out.println("The Arthematic
        exception.=)"+e.getmessage());

}
Finally{
    System.out.println("This is the
                finally block");

}}}
```

O/P:- Arthematic exception
This is the finally block.