Assignment - 2

R. Deepak
192325027.

Inheritance :

- -) Single
- -) Multilevel
- -) Multiple
- -) heirarchical
- -) hybrid

Single inheritance ;

```
class A {
      void dis A ( ) {
      System . out . println ("A"); }}
class B extends A {
        void dis B( ) {
      System . out . println ("B"); }}
class test {
      public static void main (string [] args) {
                  B obj = new B( );
                   obj . dis A ( );
                   obj . dis B ( ); }}
```

Output

```
A
B.
```

Multilevel inheritance

```
class A {
      void dis A( ) {
      System . out . println ("A") ; }}.
class B extends A {
        void dis B( ) {
      System . out . prin ln ("B") ; }}.
```

```
class c extends B {
    void disC() {
    System.out.println("C");}}
    class Main {
    public static void main(String[] args){
                    C d = new C();
                    d.disA();
                    d.disB();
                    d.disC(); }}
```

Output

```
A
B
C
```

Heirarchy

```
class A {
    void disA(){ System.out.println("A");}}
class B extends A {
    void disB(){ System.out.println("B");}
class C extends A {
    void disB() { System.out.println("C");}
class Main {
    public static void main(String[] arg
          C d = new C();
          d.disC();
          d.disA(); }}
```

Output

```
C
A.
```

Multiple

```
class A {
        int a;
         A(){
               a=5; }
        void Display A(){
               System.out.println (a); } }
interface B {
         int b = 10;
         void display B(); }
class  C extends A implemends B {
        int c;
        C(){ c=15; }
        public ~~Stat~~ void display B(){
           System.out.println (b); }
        public void display C(){
           System.out.println (c); } }
class main {
      public static void main (string[] args) {
               C d = new C();
                d.displayA();
                d.displayB();
                d.displayC(); } }
```

Output

```
5
10
15
```

Exception Handling

1) Class not found Exception.

```
public class main {
      public static void main (String [] args){
   try { class. forName ("A"); }
   catch ( ClassNotFoundException e) {
         e. printStackTrace (); }}}
```

2) Arithmetic Exception

```
public class main {
      public static void main (String [] args)
 { try { int a = 10/0;
         System.out.println (" Code in try Block")
       }
 catch ( ~~Array~~ ArithmeticException e) {
       System.out.println (" Error "); }}}.
```

3) Array Index out of Bound

```
class Main {
      public static void main (String [] args)
  { try { int a[] = {1, 2, 3};
          System.out.println (a[2]); }
  catch (ArrayIndexOutofBoundException e)
     System.out.println ("      + e.getMessage (
```

4) No such Method

```
Class Main {
        public static void main ( String [] args) {
   try { method That Does Not Exist (); }
   catch ( No such Method Exception e) {
        System.out.println(" Caught No such
                              Method Exception : " +
                              e.getMessage()); }}}
```

5) Illegal Argument Exception :

```
Class Main {
      int age ;
      public void Age ( int age ) {
           if ( age < 18 ) {
          throw new Illegal Argument Exception
                ("Age must be greate than 18"); }
         else {
            The. age = age
            System.out.printen (" + age);
         }}
   public static void main (String[] args ) {
        Person p = new person ();
        try { person . SetAge(15); }
        catch ( Illgal Argument Exception e) {
```