Assignment 6

1 Write a java program to create package and subpackage.

```
package p1;
class c1 {
        public void m1(){
                System.out.println("m1 of c1");
        }
        public static void main(String args[]) {
                c1 obj = new c1();
                obj.m1();
        }
}
package p1.p2;
class c2{
        public void m1() {
                System.out.println("m1 of c2");
        }
}
```

OUTPUT

```
m1 of c1
```

2. Write a java program to access the methods from package and sub package.

```
package pack;
public class A{
    public void msg(){
        System.out.println("Hello");
    }
}
```

```
package mypack;
import pack.*;
class B{
    public static void main(String []args){
        A obj = new A();
        obj.msg();
    }
}
```

OUTPUT

Hello

3. Exception :- CASE 1 : Problem without exception handling.

```
class A{
    public static void main(String args[]){
        int data = 100/0;
        System.out.println("data = "+data);
    }
}
```

CASE 2: Exception creates but properly handled

```
class B{
    public static void main(String args[]){
        try{
            int data = 100/0;
        }catch(Exception e){
            System.out.println("Denominator can't be zero");
        }
    }
}
```

CASE 3: Exception creates but not handled.

```
class C{
    public static void main(String []args){
        try{
            int data = 100/0;
        }catch(NullPointerException e){
            System.out.println(e);
        }
    }
}
```

~~4. Write a java program to implement Java Multi catch block.~~

```
~~class D{~~
    ~~public static void main(String[] arg~~
```

OUTPUT (CASE 1):

```
Exception in thread "main" java.lang.ArithmeticException: /by zero
        at A.main(A.java:3)
```

OUTPUT (CASE 2)

Denominator cannot be zero

OUTPUT (CASE 3)

```
Exception in thread "main" java.lang.ArithmeticException: /by zero
        at C.main(C.java:4)
```

4. Write a java program to implement Java Multicatch block.

```
class D {
    public static void main(String[] args){
        try{
            int a[] = new int[5];
            a[5] = 30/0;
        } catch (ArithmeticException e){
            System.out.println("Arithmetic Exception occurs.");
        }catch (ArrayIndexOutOfBoundException e){
            System.out.println("Array Index out of bound
                                exception occurs.");
        } catch (Exception e){
            System.out.println("Parent Exception occurs");
        }
    }
}
```

OUTPUT

Arithmetic Exception occurs.

5. Write a java program to implement Java Nested try statement.

```
public class A {
    public static void main (String args[]){
        try {
            try {
                System.out.println ("Going to divide by 0");
                int h = 5/0;
            } catch (ArithmeticException e){
                System.out.println (e);
            }
            try {
                int a[] = new int [5];
                a[5] = 10;
            }catch (ArrayIndexOutOfBoundsException e) {
                System.out.println (e);
            }catch (Exception e){
                System.out.println (e);
            }
        }
    }
}
```

OUTPUT

Going to deride by 0
java.lang.ArithmeticException: / by zero
java.lang.ArrayIndexOutOfBoundException: Index 5 out of bounds for length 5

6. Write a java program to implement the throw keyword.

```
class E {
    public static void checkAge (int age){
        if (age <18){
```

```
            throw new ArithmeticException ("Access Denied");
        }else {
            System.out.println ("Access Granted");
        }
    }
    public static void main (String [] args){
        checkAge (15);
    }
}
```

OUTPUT

~~Exception in thread "main" java.io.IOException: Device Error~~
~~at the method()~~

```
Exception in thread "main" java.lang.ArithmeticException: Access Denied
        at E.checkAge (E.java: 4)
        at E.main (E.java: 10)
```

7. Write a java program to implement the throws keyword.

→

```
import java.io.*;
class M{
    void method () throws IOException {
        throw new IOException ("device Error");
    }
}
class F{
    public static void main (String [] args) throws IOException {
        M m = new M();
        m.method ();
    }
}
```

OUTPUT

```
Exception in thread "main" java.io.IOException: device Error
        at M.method (F.java:4)
        at F.main (F.java:10)
```

8 Case1: finally example where exception doesn't occur.

```
class G{
    public static void main(String[]args){
        try{
            int data = 25/5;
            System.out.println(data);
        }catch(NullPointerException e){
            System.out.println(e);
        }finally{
            System.out.println("finally block is always
                                executed");
        }
    }
}
```

OUTPUT

```
5
finally block is always executed.
```

CASE 2: finally example where exception occurs and not handled.

```
class H{
    public static void main(String args[]){
        try{
            int data = 100/0;
            System.out.println(data);
        }catch(NullPointerException e){
```

```
            System.out.println (e);
        }finally {
            System.out.println ("finally block");
        }
    }
}
```

OUTPUT

```
finally block
Exception in thread "main" java.lang.ArithmeticException: /by zero
        at H.main (H.java:4)
```

Case 3: finally example where exception occurs and handled.

```
class I {
    public static void main (String args[]){
        try {
            int data = 100/0;
            System.out.println (data);
        }catch (ArithmeticException e){
            System.out.println ("Exception handled");
        }finally{
            System.out.println ("finally Keyword");
        }
    }
}
```

OUTPUT

```
Exception Handled
finally Keyword
```

9 Create a user-defined exception named CheckArgument to check the number of arguments passed through the command line. If the number of argument is less than 5, throw the CheckArgumentException, else print the addition of all the five numbers

```
Class CheckArgument extends Exception {
}

Class Exp {
    public static void main (String args[]){
        int i,j,s, ans=0;
        int s[] = new int [5];
        try {
            int cnt = 0;
            for (i=0 ; i< s.length ; i++) {
                s[i] = Integer. parseInt (args[i]);
                cnt ++;
                if (args .length <5) {
                    throws new CheckArgument ();
                }else if (args. length >5) {
                    throw new CheckArgument ();
                }
            }
            for (i=0 ; i< s.length ; i++){
                ans += s[i];
            }
        } catch (CheckArgument m) {
            System.out. println (" Not 5 Integer ");
        }
        System .out.println (" The sum is " + ans);
    }
```

}

~~public static void main (String args[])~~

~~{~~

OUTPUT (1)

java Exp 1 2 3 4 5

The sum is 15

OUTPUT (2)

java Exp 2 4 6 8

Not 5 Integer

The Sum is 0