LAB 6

Program 7.

Q WAP that demonstrates handling of exceptions in inheritance tree. Create a base class "father" and derived class called "son" which extends the base class. In father class a constructor which takes the age & throws the exception WrongAge() when input age <0. In Son class, implement a constructor that takes both father & son's age & throw an exception if Son's age is >= father's age.

```
import java.util.*;
class WrongAge extends Exception
{
  String message;
  public WrongAge (String msg)
  { this.message = msg;
    System.out.println(msg);
  }
}

class Father throws WrongAge
{
  int age;
  Father (int fage)
  {
    this.age = fage;
    if (fage < 0)
    {
      throw new WrongAge ("Age cant be less than zero");
    }
```

```
int getAge()
{
  return age;
}

class Son extends Father throws WrongAge
{
  int s_age;
  Son(int f_age, int s_age)
  {
    super(f_age); this.s_age = s_age;
    if(f_age <= s_age)
    {
      throw new WrongAge("Age of son cant be greater than father's age.");
    }
  }
  int getSAge()
  {
    return s_age;
  }
}.

class Main
{
  public static void main(String args[])
  { int n1, n2;
    Scanner input = new Scanner(System.in);
    System.out.println("Enter father's age:");
    n1 = input.nextInt();
    System.out.println("Enter Son's age:");
    n2 = input.nextInt();
```

```
try{
  Father obj1 = new father(n1)
  Son obj2 = new son(n1, n2)
  }
catch (^WrongAge e)
  {
    System.out.println ("caught"); }.
}
```

~~System.out.println (obj1.getAge())~~

```
System.out.println ("Father's age is:" +
  obj1.getAge());
System.out.println ("Son's age is:" +
  obj2.getSAge());
}
```

OUTPUT:

Enter Father age: 0 Enter Sons age: 10
Father cant be younger than son
caught

Enter Father age: 40 Enter Sons age: 20

Program 8

Q WAP which creates two threads, one thread displaying "BMS college Engineering" once every ten seconds & another displaying "CSE" once every 2 seconds.

```
import java.util.*;
class Thread1 extends Thread
{
    public void run()
    { while(true)
      { System.out.println("BMS college of
Engineering");
      }
    try {
          Thread.sleep(10000);
        }
    catch (InterruptedException e)
    {
      System.out.println("Caught");
    }
  }}

class Thread2 extends Thread
{
    public void run()
    { while(true)
      { System.out.println("CSE");
      }
      try{
            Thread.sleep(2000);
          }
      catch (InterruptedException e)
```

```
        {
            System.out.println("caught");
        }
    }
}

class Main
{
    public static void main (String args[])
    {
        Thread1 t1 = new Thread1();
        Thread2 t2 = new Thread2();

        t1.start();
        t2.start();
    }
}
```

Output:

```
CSE
BMS college of Engineering
 CSE
 CSE
 CSE
 CSE
BMSCE
BMSCE
BMSCE
BMSCE
```

Algorithm for Q 7.

step 1: start

step 2: Create a class WrongAge extending Exception
prints a String msg.

step 3: ~~class~~ create class father throws exception WrongAge
create a parameterized constructor
if f_age < 0 then throw exception WrongAge

Step 4: create class son which extends father, throws exception WrongAge
create a parameterized constructor
if (f_age <= s_age then throw exception WrongAge

Step 5: create a class Main.
Input two variables n1, n2.
In try block create an object obj1 for class father.
create an object obj2 for class son
catch ~~&~~ WrongAge exception.

Step 6: stop.

Algorithm for Program 8:

Step 1: start
Step 2: create a class Thread1 extending Thread.
Step 3: In the method run
while (true)
Print BMS College of Engineering
let the thread sleep for 10seconds
end of while.
catch & Interrupted Exception and
Step 4: print caught.

Step 4: create a class Thread2 extending Thread.
Step 5: In the method run
while (true)
print CSE
let the thread sleep for 2seconds
end of while.
catch any Interrupted Exception and
print caught.

Step 6: create a class Main
create a Thread t1 for class Thread1
create a Thread t2 for class Thread2.
start t1.
Start t2.
Step 7: Stop.

16.02.24

```
C:\Users\BMSCE\Desktop\1BM22CS012\lab7,8>javac Main.java

C:\Users\BMSCE\Desktop\1BM22CS012\lab7,8>java Main
Enter Fathers age
0
Enter Sons age
10
Father cant be younger than son
Caught
Name:Adikar Charvi USN:1BM22CS012

C:\Users\BMSCE\Desktop\1BM22CS012\lab7,8>javac Main.java

C:\Users\BMSCE\Desktop\1BM22CS012\lab7,8>java Main
Enter Fathers age
-25
Enter Sons age
12
Age cant be less than 0
Caught
Name:Adikar Charvi USN:1BM22CS012

C:\Users\BMSCE\Desktop\1BM22CS012\lab7,8>javac Main.java

C:\Users\BMSCE\Desktop\1BM22CS012\lab7,8>java Main
Enter Fathers age
20
Enter Sons age
12
Name:Adikar Charvi USN:1BM22CS012
```

```
C:\Users\BMSCE\Desktop\1BM22CS012\lab7,8>javac TDemo.java

C:\Users\BMSCE\Desktop\1BM22CS012\lab7,8>java TDemo
Name:Adikar Charvi USN:1BM22CS012
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
BMS College of Engineering
BMS College of Engineering
BMS College of Engineering
BMS College of Engineering

C:\Users\BMSCE\Desktop\1BM22CS012\lab7,8>
```