Date 16/02/2024
Page

Week - 5

1) CIE package

```
package CIE;
public class Personal {
    public String usn;
    public String name;
    public int sem;
    public Personal (String usn, String name, int sem){
    this.usn = usn;
    this.name = name;
    this.sem = sem;
    }
}

Import java.util.Arrays;
public class Internals {
    public int[] InternalMarks;
    public Internals (int[] InternalMarks){
    this.InternalMarks = InternalMarks;
    }
}

package SEE;
Import CIE.Personal;
public class External extends Personal {
    public int[] SeeMarks;
    public External (String num, String name, int sem,
int[] seeMarks)
    Super (usn, name, sem);
    this.SeeMarks = SeeMarks;
}
}
```

2024

Date __/__/__
Page _____

```
package FormMain;
import java.util.Arrays;
import CIE.Internals;
import CIE.Personal;
import SEE.External;
public class Main {
    public static void main(String[] args) {
    int n=3;
    Student[] students = new Student[n];
    for(int i=0; i<n; i++) {
    int[] InternalMarks = {80, 75, 90, 85, 95};
    int[] seeMarks = {30, 80, 75, 90, 85};
    students[i] = new Student(new Personal("USN"+i,
     "Student"+i, i), new Internal(InternalMarks));
    students[i].see = new External("USN"+i, "Student"+i, i,
                          seeMarks);
    for(int i=0; i< students.length; i++) {
      Student student = students[i];
      SOP("Student: "+ student.personal.name);
      SOP("Internal Marks" + Arrays.toString(Student.internals.
InternalMarks));
      SOP("SEE Marks " + Arrays.toString(Student.see.seeMarks));
      SOP();
    }
}.

Static class Student {
    public Personal personal;
    public Internals internals;
    public External see;
    public Student(Personal personal, Internals internals) {
        this.personal = personal;
        this.internals = internals;
    } }.
```

Date __/__/__
Page _____

output:

Student. Student 0
Internal Marks [80, 75, 90, 85, 95]
SEE Marks [70, 80, 75, 90, 85]

Student Student 1
Internal Marks: [80, 75, 90, 85, 95]
SEE Marks: [70, 80, 75, 90, 85]

Student. student 2
Internal Marks: [80, 75, 90, 85, 95]
SEE Marks: [70, 80, 75, 90, 85].

Date / /
Page

Week-6

2) Write a program that demonstrates handling of exceptions. In Inheritience tree. Create base class called Father and derived class called son which extends the base class. In Father class, Implement a constructor which takes the age and throws the exception WrongAge(), when the input age<0. In son class, Implement a constructor that takes both father and son's age and throws an exception if son's age is >= father's age.

```
class Father {
  public int age;
  Father (int age) {
   if (age>0) {
     throw new IllegalArgumentException ("Age cannot be -ve");
    }.
   this.age = age;
   }
}.
public class Son extends Father {
    public int SonAge;
    public Son (int fatherage, int SonAge) {
    Super (fatherAge).
    if ( SonAge >= fatherAge) {
      throw new IllegalArgument Exception ("son age cannot be
                                    >= father age");
    }.
    this. SonAge = SonAge;
   }
}
```

```
import java.util.Scanner;
public class Main {
    public static void Main (String[] args) {
        Scanner s = new Scanner (System.in);
        try {
            SOP ("Enter father age:");
            Int fatherAge = S.NextInt();
            SOP ("Enter Son's age:");
            Int SonAge = S.nextInt();
            Son son = new son (fatherAge, SonAge);
            SOP ("Father age:" + son.age);
            SOP ("Son's age:" + son.SonAge);
        }
        catch (Illegal Argument Exception e) {
            Sop ("Exception:" + e.getMessage());
        }
        s.close();
    }
}
```

Output:

Enter Father's age: 20
Enter Son's age: 40
Exception: Son's age cannot be >= father's age.

Date __/__/__
Page _____

3) Write a program which creates two threads, one thread displaying "BMS college of Engineering" once every ten seconds and another displaying "CSE" once every two seconds.

```
class BMSDisplay extends Thread {
    public void run() {
        while (true) {
            System.out.println("BMS college of Engineering");
            try {
                Thread.sleep(10000);
            }
            catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}

class CSEDisplay extends Thread {
    public void run() {
        while (true) {
            SOP("CSE");
            try {
                Thread.sleep(2000);
            } catch (InterruptedException e).
                e.printStackTrace();
            }
        }
    }
}
```

```
public class Main {
    public static void main (String[] args) {
        BMSDisplay bmsThread = new BMSDisplay();
        CSEDisplay cseThread = new CSEDisplay();

        bmsThread.start();
        cseThread.start();
    }
}
```

Output:

```
BMS College of Engineering
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
```