

Week - 5

1) CIE package

```
package CIE;

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

Import java.util. Arrays;

```
public class Internal {
    public int[] internalMark;
    public Internal (int[] internalMark) {
        this.internalMark = internalMark;
    }
}
```

package SEE;

Import CIE. Person;

```
public class External extends Person {
    public int[] seeMark;
    public External (String usn, String name, int sem,
        int[] seeMark) {
        super (usn, name, sem);
        this.seeMark = seeMark;
    }
}
```

```

package FormMain;
import java.util. Array;
import CIE. Internal;
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[] InternalMarkk = { 80, 75, 90, 85, 95 };
            int[] seeMarkk = { 30, 80, 75, 90, 85 };
            students[i] = new Student (new Personal ("USN"+i,
                "Student"+i, 1), new Internal (InternalMarkk));
            students[i].see = new External ("USN"+i, "Student"+i, 1,
                seeMarkk);
        }
        for (int i=0; i<students.length; i++) {
            Student student = students[i];
            sop ("Student: " + student.personal.name);
            sop ("Internal Markk" + Array to string (student.internal.
                InternalMarkk));
            sop ("SEE Markk " + Array to string (student.see.seeMarkk));
            sop ();
        }
    }
}

```

1st Sem,

```

static class Student {
    public Personal personal;
    public Internal internal;
    public External see;
    public Student (personal personal, Internal internal) {
        this.personal = personal;
        this.internal = internal;
    }
}

```

output:

Student Student n

Internal Markk [80, 75, 90, 85, 95]

SFE Markk [70, 80, 75, 90, 85]

Student Student1

Internal Markk: [80, 75, 90, 85, 95]

SFE Markk: [70, 80, 75, 90, 85]

Student Student2

Internal Markk: [80, 75, 90, 85, 95]

SFE Markk: [70, 80, 75, 90, 85]

Week-6

- 2) Write a program that demonstrates handling of exceptions. In inheritance 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 Wrong Age(), 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 \geq 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 IllegalArgumentException ("son age cannot be
            >= father age");
        }
        this.SonAge = SonAge;
    }
}
```

3-

```

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 (IllegalArgument 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.

- 3) Write a program which creates two threads, one thread displaying "BMS collage 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 collage 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 bmgThread = new BMSDisplay();  
        CSEDisplay cseThread = new CSEDisplay();  
  
        bmgThread.start();  
        cseThread.start();  
    }  
}
```

Output:

BMS Collage of Engineering

CSE

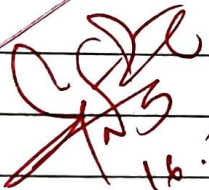
CSE

BMS Collage of Engineering

CSE

CSE

CSE


16.02.24