

LAB 07 : EXPECTATION

LAB-07 :-

26/01/23
Page

8. WAP that demonstrates handling of exceptions in java. create base class called "Father" in Father class implement a constructor which takes the age & throws the exception WrongAge() when the input age < 0.

```
import java.util.Scanner;

class WrongSalary extends Exception {
    public WrongSalary() {
        super("salary error");
    }
    public WrongSalary(String message) {
        super(message);
    }
}

class InputScanner {
    public static int getInput(String prompt) {
        Scanner scanner = new Scanner(System.in);
        System.out.print(prompt);
        return scanner.nextInt();
    }
}

class Father extends InputScanner {
    public int fatherAge;
```

```

class son extends father {
    private int sonAge;

    public son() throws WrongAge {
        super();
        sonAge = get Input ("Enter son's age:");
        if (sonAge > super.fatherAge) {
            throw new WrongAge ("son's age cannot be
            greater than father's age");
        } else if (sonAge < 0) {
            throw new WrongAge ("Age cannot be negative");
        }
        else if (sonAge == super.fatherAge)
        { throw new WrongAge ("son age cannot be equal
        to father's age !!!");
        }
    }

    public void display() {
        super.display();
        System.out.println("son's Age : " + sonAge);
    }
}

```

```

public class Exception Handling {
    public static void main (String args[]) {
        try {
            son son = new son();
            son.display();
        } catch (WrongAge e) {
            System.out.println ("Error : " + e.getMessage());
        }
    }
}

```

output:

(i) Enter Father's age : 54
Enter Son's age : 19
Father age : 54
Son's age : 19

(ii) Enter Father's age : 19
Enter Son's age : 54
Error : Son's age cannot be greater than father's age

(iii) Enter Father's age : -2
Error : Age cannot be -ve

(iv) Enter Father's age : 12
Enter Son's age : 12
Error : Son's age cannot be equal to father

(v) Enter Father's age : 45
Enter Son's age : -3
Error : Age cannot be -ve

24.02.24

Code :

```
import java.util.Scanner;

class WrongAge extends Exception {

    public WrongAge() {

        super("Age Error");

    }

    public WrongAge(String message) {

        super(message);

    }

}

class InputScanner {

    public static int getIntInput(String prompt) {

        Scanner scanner = new Scanner(System.in);

        System.out.print(prompt);

        return scanner.nextInt();

    }

}

class Father extends InputScanner {

    public int fatherAge;

    public Father() throws WrongAge {

        fatherAge = getIntInput("Enter Father's age: ");

        if (fatherAge < 0) {
```

```

        throw new WrongAge("Age cannot be negative");
    }
}

public void display() {
    System.out.println("Father's Age: " + fatherAge);
}
}

class Son extends Father {
    private int sonAge;

    public Son() throws WrongAge {
        super();
        sonAge = getIntInput("Enter Son's age: ");
        if (sonAge > super.fatherAge) {
            throw new WrongAge("Son's age cannot be greater than father's age");
        } else if (sonAge < 0) {
            throw new WrongAge("Age cannot be negative");
        }
        else if(sonAge==super.fatherAge)
        { throw new WrongAge("Son age cannot be equal to father's age!!!!");
        }
    }

    public void display() {

```

```

        super.display();

        System.out.println("Son's Age: " + sonAge);

    }

}

public class ExceptionHandling {

    public static void main(String args[]) {

        try {

            Son son = new Son();

            son.display();

        } catch (WrongAge e) {

            System.err.println("Error: " + e.getMessage());

        }

    }

}

```

Additional prgm :

```

import java.util.Scanner;

class WrongSalary extends Exception {

    public WrongSalary() {

        super("Salary Error");

    }

    public WrongSalary(String message) {

        super(message);

    }

}

class InputScanner {

    public static int getIntInput(String prompt) {

```

```

        Scanner scanner = new Scanner(System.in);
        System.out.print(prompt);
        return scanner.nextInt();
    }
}

class Manager extends InputScanner {
    public int ManagerSalary;

    public Manager() throws WrongSalary {
        ManagerSalary = getIntInput("Enter Manager Salary: ");
        if (ManagerSalary < 0) {
            throw new WrongSalary("Salary cannot be negative");
        }
    }

    public void display() {
        System.out.println("Manager Salary: " + ManagerSalary);
    }
}

class Emp extends Manager {
    private int EmpSalary;

    public Emp() throws WrongSalary {
        super();
        EmpSalary = getIntInput("Enter Employee age: ");
        if (EmpSalary > super.ManagerSalary) {
            throw new WrongSalary("Employee Salary cannot be greater than
Manager salary ");
        } else if (EmpSalary < 0) {
            throw new WrongSalary("Salary cannot be negative");
        }
        else if(EmpSalary==super.ManagerSalary)
        { throw new WrongSalary("Salary cannot be equal to Manager salary!!!!");
        }

    }

    public void display() {
        super.display();
    }
}

```

```
        System.out.println("employee salary: " + EmpSalary);
    }
}

public class ExceptionHandlingdemo {
    public static void main(String args[]) {
        try {
            Emp emp = new Emp();
            emp.display();
        } catch (WrongSalary e) {
            System.err.println("Error: " + e.getMessage());
        }
    }
}
```