| AIM: | To solve and modify our previous programs using exception handling. |
|---|---|
| **Program 1** | |
| **PROBLEM STATEMENT :** | **Define a class Cricketer which has:-**<br>**Attributes:-**<br>**• player_name**<br>**• runs_hit**<br>**• innings_count**<br>**• not_out_count**<br>**• batting_avg**<br>**Methods:-get_avg**<br>**Make a cricket team with 11 cricketers. For each cricketer, find his batting average. Handle all different errors while calculating this. Also, make a method which will find the list of cricketers in ascending order of their batting average and also display the cricketer stats in this order.**<br>**If the average of the batting average of the entire team is less than 20 runs then throw a user-defined exception.**<br>**Note- handle errors like ArrayIndexOutOfBoundsException, ArithmeticException,ArrayStoreException, NumberFormatException, etc** |
| **PROGRAM:** | import java.util.Scanner;<br>class lessRuns extends Exception{<br>lessRuns()<br>{<br>super();<br>}<br>}<br>class Cricketer {<br>String player_name;<br>int runs_hit;<br>int innings_count;<br>int not_out_count;<br>double batting_avg;<br>Scanner sc = new Scanner(System.in);<br><br>void get_avg() {<br>try {<br>batting_avg = (double) (runs_hit / innings_count);<br>} catch (ArithmeticException e) {<br>System.out.println("What is this. you are trying to divide by 0");<br>} |

```
    }

    void getData() {
        System.out.print("Enter player name: ");
        player_name = sc.next();
        System.out.print("Enter runs hit: ");
        runs_hit = sc.nextInt();
        System.out.print("Enter number of innings played: ");
        innings_count = sc.nextInt();
        System.out.print("Enter number of not out innings played: ");
        not_out_count = sc.nextInt();
    }
}

public class testcricket {
    public static void main(String[] args) {
        Cricketer obj[] = new Cricketer[11];
        try {
            for (int i = 0; i < 2; i++) {
                try {
                    System.out.println("For player " + (i + 1));
                    obj[i] = new Cricketer();
                    obj[i].getData();
                    obj[i].get_avg();
                } catch (ArrayStoreException m) {
                    System.out.println(m + " handled.");
                }
            }
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println(e + " handled.");
        }
        double entireAvg=0;
        for (int i = 0; i < 2; i++) {
            entireAvg += obj[i].batting_avg;
        }
        entireAvg/=2;
        try{
            check(entireAvg);
        }
        catch(Exception e)
        {
            System.out.println("The team has a batting average of less than 20 runs.");
        }
        Cricketer temp = new Cricketer();
        for(int i = 0; i < 2 ; i++)
        {
```

```
        for(int j = i+1 ; j < 2 ; j++)
        {
           if(obj[j].batting_avg<obj[i].batting_avg)
           {
              temp = obj[j];
              obj[j] = obj[i];
              obj[i] = temp;
           }
        }
     }
     for(int i = 0; i < 2; i++)
     {

System.out.print(obj[i].player_name+"\t"+obj[i].runs_hit+"\t"+obj[i].innings_count+"\t" +
obj[i].not_out_count+"\t" + obj[i].batting_avg+"\n");
     }
  }
  static void check(double x) throws lessRuns{
     if(x<20)
     {
        throw new lessRuns();
     }
  }
}
```

**RESULT:**

```
For player 1
Enter player name: r
Enter runs hit: 23
Enter number of innings played: 1
Enter number of not out innings played: 1
For player 2
Enter player name: q
Enter runs hit: 45
Enter number of innings played: 3
Enter number of not out innings played: 2
The team has a batting average of less than 20 runs.
q    45  3    2    15.0
r    23  1    1    23.0
```

| Program 2 | |
|---|---|
| **PROBLEM STATEMENT :** | **Write a program to accept distance between two vaccine dose from 1-84 as input from user. If the user enters <84 days as an input or if user enters any negative number, or >100 user defined exception should be generated.** |
| **PROGRAM:** | |

```
import java.util.Scanner;
class InvalidAgeException extends Exception{
        InvalidAgeException(String s)
        {
         super(s);
        }
}
class vaccine{
        public static void  main(String[] args){
                Scanner sc=new Scanner(System.in);
                System.out.println("Enter the distance");
                int distance=sc.nextInt();
                try{
                        validate(distance);
                }
                catch(Exception m){System.out.println(": "+m);}

        }
        static void validate(int distance) throws InvalidAgeException {
                if(distance<=84||distance>=100) throw new
InvalidAgeException("Arre sir kya likh rahe ho, WORLD RECORD bananna hai
kya");
                else
                        System.out.println("Covid 19 tikakaran yaha uplabdh
hai");
                }
}
```

**RESULT:**

```
Enter the distance: 56
InvalidAgeException: Arre sir kya likh rahe ho, WORLD RECORD bananna hai kya
```

```
Enter the distance: 89
Covid 19 tikakaran yaha uplabdh hai
```

| Program 3 | |
|---|---|
| **PROBLEM STATEMENT:** | **There is an abstract class Account**<br>**Attribute:-**<br>**• Name**<br>**• Balance**<br>**• Acc_No**<br>**Method:-**<br>**• Deposit - abstract method** |

| | |
|---|---|
| | **• withdraw - abstract method**<br>**• display - abstract method**<br>**Saving Account inherits the Account class and provides the implementation for the methods accordingly**<br>**Saving Account class Attribute:-**<br>**• interestRate**<br>**• minBalance**<br>**Method**<br>**• addInterest: handle Arithmetic Exception**<br>**• transfer():**<br>**Note:**<br>**• Balance cannot be less than 0.**<br>**• In a Saving account if minBalance is set then for that the balance cannot go less than that amount. If it goes, an error must be shown.**<br>**• let the user deposit to or withdraw from the account. For each transaction, a message is displayed to indicate the status of the transaction: successful or failed. In case of failure, the failure reason is reported.**<br>**• The possible Exceptions are negative-amount-exception (in both deposit and withdraw transaction) and insufficient-amount-exception ( in withdraw transaction).**<br>**For the above scenario write an interactive program in Java. Also, show output for different use cases.** |
| **PROGRAM:** | import jdk.swing.interop.SwingInterOpUtils;<br><br>import java.util.*;<br>abstract class Account{<br>String name;<br>double balance;<br>long Acc_No;<br>abstract void deposit(double amount);<br>abstract void withdraw(double amount);<br>abstract void chk_balance();<br>}<br>class Saving_Acc extends Account{<br>private double interestRate;<br>static double SI =0;<br>double minibalance=25000;<br>Saving_Acc(String name,double amount){<br>this.name=name; |

```
        balance=amount;
        interestRate=0.3;
    }
    void deposit(double amount){
        balance+=amount;
        System.out.println("Done");
    }
    void withdraw(double amount){
        try{
            check(amount);
            balance-=amount;
            System.out.println("Done");
        }
        catch (Exception e){
            System.out.println(e);
        }
    }
    void check (double amount)throws insuf_amoun_exp{
        if ((balance-amount)<minibalance){
            throw new insuf_amoun_exp("Bhai paisa ka hai tera paas");
        }
    }
    void chk_balance(){
        System.out.println("Abi aapake pass "+ balance+" Rupee hai. kya karnge etni dhanrashi
ka");
    }
    public void addInterest ( int T){
        try
        {
            SI += (minibalance * T * interestRate) / 100;
        }
        catch(ArithmeticException e){}
    }
    class insuf_amoun_exp extends Exception{
        insuf_amoun_exp(String s){
            super(s);
        }
    }
}
public class Bank {
    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter the name of customer: ");
        String name=sc.nextLine();
        System.out.print("Enter the Balance: ");
        double balance=sc.nextDouble();
```

```
        Saving_Acc p1= new Saving_Acc(name,balance);
        boolean flag=true;
        while(flag){

System.out.println("==========================================================
========================");
            System.out.println("aapake paas 6 vikalp hain jinhen aap chunenge");
            System.out.println("Option1: Deposit\nOption2: Withdraw\nOption3: Find
Interest\nOption4: Check your Balance\nOption5:Exit");
            System.out.print("Enter Option : ");
            int option=sc.nextInt();
            switch(option){
               case 1:
                  System.out.print("Raashi Daalen : ");
                  int amount=sc.nextInt();
                  p1.deposit(amount);
                  break;

               case 2:
                  System.out.print("Raashi Daalen : ");
                  int amount2=sc.nextInt();
                  p1.withdraw(amount2);
                  break;
               case 3:
                  System.out.print("Enter no of years : ");
                  int tim2e=sc.nextInt();
                  p1.addInterest(tim2e);
                  break;
               case 4:
                  p1.chk_balance();
                  break;
               case 5:
                  flag=false;
                  System.out.println("Dhanyavaad, Ham aasha karate hain ki aapaka din shubh ho");

System.out.println("==========================================================
========================");
                  break;
               default:
                  System.out.println("Invalid option");
                  break;
            }
        }
    }
}
```

**RESULT:**

```
Enter the name of customer: teri
Enter the Balance: 30000
================================================================================
aapake paas 6 vikalp hain jinhen aap chunenge
Option1: Deposit
Option2: Withdraw
Option3: Find Interest
Option4: Check your Balance
Option5:Exit
Enter Option : 1
Raashi Daalen : 20000
Done
================================================================================
aapake paas 6 vikalp hain jinhen aap chunenge
Option1: Deposit
Option2: Withdraw
Option3: Find Interest
Option4: Check your Balance
Option5:Exit
Enter Option : 2
Raashi Daalen : 4000
Done
================================================================================
aapake paas 6 vikalp hain jinhen aap chunenge
Option1: Deposit
Option2: Withdraw
Option3: Find Interest
Option4: Check your Balance
Option5:Exit
Enter Option : 3
Enter no of years : 2
================================================================================
aapake paas 6 vikalp hain jinhen aap chunenge
Option1: Deposit
Option2: Withdraw
Option3: Find Interest
Option4: Check your Balance
Option5:Exit
Enter Option : 4
Abi aapake pass 46000.0 Rupee hai. kya karnge etni dhanrashi ka
================================================================================
aapake paas 6 vikalp hain jinhen aap chunenge
Option1: Deposit
Option2: Withdraw
Option3: Find Interest
Option4: Check your Balance
Option5:Exit
Enter Option : 5
Dhanyavaad, Ham aasha karate hain ki aapaka din shubh ho
================================================================================
```

| | |
|---|---|
| **CONCLUSION:** | We learnt about exception handling and solved some problems using that. |