Name	**Virinchi Sadashiv Shettigar**
UID no.	**2021300118**
Experiment No.	**9**

AIM:	Implement a program to demonstrate Exception Handling.
Program 1	
PROBLEM STATEMENT:	**Define a class Cricketer which has:-** **Attributes:-** **● player_name** **● runs_hit** **● innings_count** **● not_out_count** **● batting_avg** **Methods:-get_avg** **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.** **If the average of the batting average of the entire team is less than 20 runs then throw a user-defined exception.** **Note- handle errors like ArrayIndexOutOfBoundsException, ArithmeticException,ArrayStoreException, NumberFormatException, etc**

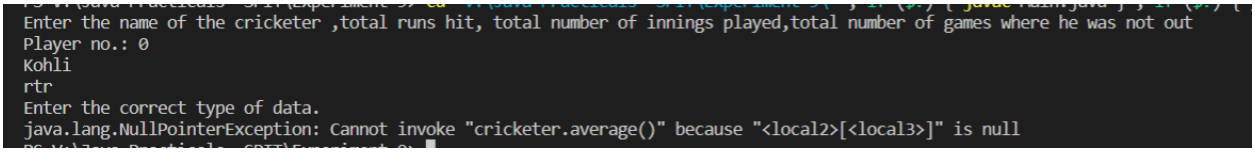
PROGRAM:

```
import java.util.InputMismatchException;
import java.util.Scanner;

class cricketer {
    String playerName = "";
    int runsHit = 0, inningsCount = 0, notOutCount = 0;
    int battingAvg = 0, flag = 0;

    cricketer() {

    }

    cricketer(String playerName, int runsHit, int inningsCount, int
notOutCount) {

        this.playerName = playerName;
        this.runsHit = runsHit;
        this.inningsCount = inningsCount;

        this.notOutCount = notOutCount;
    }

    public void average() {
        try {
            battingAvg = (int) runsHit / inningsCount;
            if (battingAvg < 20) {
                throw new ArithmeticException("Average is too low for a player");
            }
        } catch (ArithmeticException e) {
            System.out.println("player hasnt't played a single game so data is
invalid");
            flag = 1;
        }

    }

    public void display() {
        System.out.println("The data of cricketeres are as follows: \n");
        System.out.println("The name of the player is: " + playerName);
        System.out.println("The total runs hit by the player is : " + runsHit);
        System.out.println("The number of innings played by the player is: " +
inningsCount);
        System.out.println("The number of total not out innings of the player
```

```
is: " + notOutCount);
        System.out.println("The batting average of the player is: " +
battingAvg);
    }
}

public class Main {
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        cricketer[] c = new cricketer[3];
        try {
            for (int i = 0; i < 3; i++) {
                System.out.println(
                        "Enter the name of the cricketer ,total runs hit, total number
of innings played,total number of games where he was not out \nPlayer
no.: " + i);

                try {
                    c[i] = new cricketer(sc.nextLine(), sc.nextInt(), sc.nextInt(),
sc.nextInt());
                    sc.nextLine();
                } catch (Exception t) {
                    System.out.println("Enter the correct type of data.");
                }
                c[i].average();
            }
            for (int i = 0; i < 2; i++) {
                for (int j = i; j < 3; j++) {
                    if (c[i].battingAvg > c[j].battingAvg) {
                        cricketer ob = new cricketer();
                        ob = c[i];
                        c[i] = c[j];
                        c[j] = ob;
                    }
                }
            }
            for (int i = 0; i < 3; i++) {
                if (c[i].flag == 0) {
                    c[i].display();
                }
            }
            sc.close();
```

```
        } catch (NullPointerException g) {
            System.out.println(g);
        }
    }
}
```

RESULT:



Program 2

PROBLEM STATEMENT:

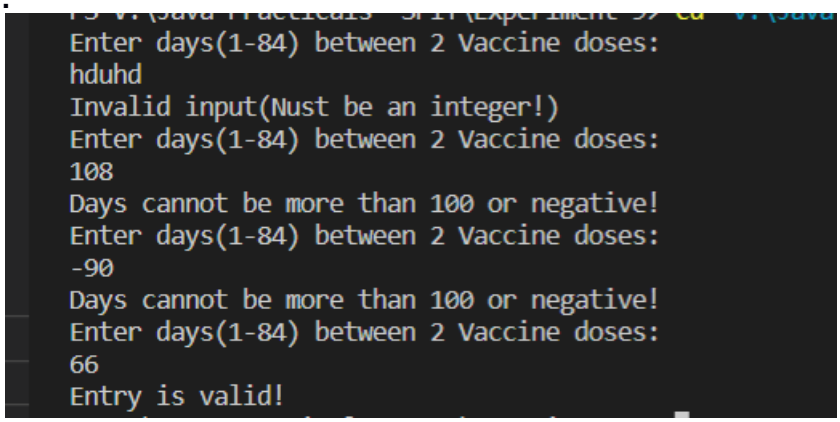
Write a program to accept distance between two vaccine dose from 1-84 as input from user. If the user enters <84 days as an input or if user enters any negative number, or >100 user defined exception should be generated.

PROGRAM:

```
 import java.util.*;
class MyException extends Exception {
    public MyException(int days) {
        super();
    }
}
public class Vaccine {
    public static void main(String[] args) {
        int days,flag=0;
        Scanner sc = new Scanner(System.in);
        while(flag==0) {
            System.out.println("Enter days(1-84) between 2 Vaccine doses:");
            try {
                days = sc.nextInt();
                if(days>100 || days<0) {
                    throw new MyException(days);
                }
                else {
                    flag = 1;
                }
```

```
            System.out.println("Entry is valid!");
        }
        catch (InputMismatchException e) {
            System.out.println("Invalid input(Nust be an integer!)");
            sc.nextLine();
            flag=0;
        }
        catch (MyException ex) {
            System.out.println("Days cannot be more than 100 or negative!");
            flag=0;
        }
    }
    sc.close();
  }
}
```

RESULT:



Program 3

PROBLEM STATEMENT:

There is an abstract class Account

Attribute:-

- **Name**
- **Balance**
- **Acc_No**

Method:-

- **Deposit - abstract method**
- **withdraw - abstract method**
- **display - abstract method**

	Saving Account inherits the Account class and provides the implementation for the methods accordingly **Saving Account class Attribute:-** **● interestRate** **● minBalance** **Method** **● addInterest: handle Arithmetic Exception** **● transfer():** **Note:** **● Balance cannot be less than 0.** **● 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.** **● 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.** **● The possible Exceptions are negative-amount-exception (in both deposit and withdraw transaction) and insufficient-amount-exception (in withdraw transaction).** **For the above scenario write an interactive program in Java. Also, show output for different use cases.**

PROGRAM:	import java.util.Scanner;
	class insufficientAmountException extends Exception {
	  insufficientAmountException(String s) {
	  super(s);
	  }
	}
	
	class negativeBalanceException extends Exception {
	  negativeBalanceException(String s) {
	  super(s);
	  }
	}

```
abstract class account {
    Scanner sc = new Scanner(System.in);
    String name;
    int balance = 5000;
    int accNo;

    account() {
        name = sc.nextLine();
        accNo = sc.nextInt();
        sc.nextLine();
    }

    abstract void deposit(int amount);

    abstract void withdraw(int amount) throws insufficientAmountException;

    abstract void display();
}


class SavingsAccount extends account {
    int minimumBalance = 5000;
    double intrest = 3.5;

    SavingsAccount() {
        super();
    }

    public void deposit(int amount) {
        balance += amount;
    }

    @Override
    public void withdraw(int amount) throws insufficientAmountException {
        if (balance - amount < 5000) {
            throw new insufficientAmountException("");
        } else {
            balance -= amount;
        }
    }

    @Override
    void display() {
```

```
        System.out.println("The amount in the account is :" + balance);
    }

    public void addIntrest() {
        balance += intrest * balance / 100;

    }

    public void transfer(int amount) throws insufficientAmountException {
        if (balance - amount < 5000) {
            throw new insufficientAmountException("");
        } else {
            balance -= amount;
        }
    }
}

public class ExceptionHandling {
    static void validateAmount(int amount) throws
negativeBalanceException {
        if (amount < 0) {
            throw new negativeBalanceException("");
        }
    }

    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter your name and your account number");
        SavingsAccount person = new SavingsAccount();
        int flag = 1;
        while (flag == 1) {
            System.out.println("Enter \n1) If you want to add money in your
account \n2) If you want to withdraw money from your account\n3) If you
want to display the money in youraccount\n4) If you want to receive intrest
on your account\n5) If you want to send money to someonefrom your
account ");
            int option = s.nextInt();
            switch (option) {
                case 1:

                    System.out.println("enter the amount you wish to add to your
bank account");
```

```
                try {
                    int deposits = s.nextInt();
                    validateAmount(deposits);
                    person.deposit(deposits);
                    person.display();
                    break;
                } catch (negativeBalanceException n) {
                    System.out.println(n + "enter a valid amount");
                }
            case 2:
                System.out.println("Enter the amount you wish to
withdrawfrom your bank account ");
                int amount = s.nextInt();
                try {
                    validateAmount(amount);
                    try {

                        person.withdraw(amount);
                        person.display();
                        break;
                    } catch (insufficientAmountException t) {
                        System.out.println(t + "Sorry transaction is not possible
due to insufficient balance");
                        break;
                    }

                } catch (negativeBalanceException e) {
                    System.out.println(e + "Enter A positive amount");
                    break;
                }
            case 3:
                System.out.print("The balance in your bank account is: ");
                person.display();
                break;
            case 4:
                person.addIntrest();
                person.display();
                break;
            case 5:
                System.out.println("Enter the amount you wish to transfer");
                int a = s.nextInt();
                try {
                    validateAmount(a);
```

```
                    try {
                        person.transfer(a);
                        person.display();
                        break;
                    } catch (insufficientAmountException t) {
                        System.out.println(t + "sorry transaction is not possible
due to insufficient balance");
                        break;
                    }
                }

                catch (negativeBalanceException e) {
                    System.out.println(e + "Enter A positive amount");
                    break;
                }
            default:
                try {
                    throw new ArithmeticException("");
                } catch (ArithmeticException e) {
                    System.out.println(e + " Enter a valid number");
                    break;
                }
            }

            System.out.println("Enter 1 if you want to continue the program else
press 0");
            flag = s.nextInt();
        }
        s.close();
    }
}
```

RESULT:

```
Enter your name and your account number
Hello
3424
Enter
1) If you want to add money in your account
2) If you want to withdraw money from your account
3) If you want to display the money in youraccount
4) If you want to receive intrest on your account
The balance in your bank account is: The amount in the account is :5000
Enter 1 if you want to continue the program else press 0
1
Enter
1) If you want to add money in your account
2) If you want to withdraw money from your account
3) If you want to display the money in youraccount
4) If you want to receive intrest on your account
5) If you want to send money to someonefrom your account
1
enter the amount you wish to add to your bank account
200
The amount in the account is :5200
Enter 1 if you want to continue the program else press 0
1
Enter
1) If you want to add money in your account
2) If you want to withdraw money from your account
3) If you want to display the money in youraccount
4) If you want to receive intrest on your account
5) If you want to send money to someonefrom your account
4
The amount in the account is :5382
Enter 1 if you want to continue the program else press 0
1
Enter
1) If you want to add money in your account
2) If you want to withdraw money from your account
3) If you want to display the money in youraccount
```

```
4) If you want to receive intrest on your account
5) If you want to send money to someonefrom your account
2
Enter the amount you wish to withdrawfrom your bank account
6000
insufficientAmountException: Sorry transaction is not possible due to insufficient balance
Enter 1 if you want to continue the program else press 0
0
```

CONCLUSION:	In this experiment, we learned about exception handling in java and how can you create your own exception.