NAME: ADWAIT S PURAO

UID: 2021300101

BATCH: B2

Q1:

Program 1:

Write a java program to throw a exception (checked) for an employee details. a) If an employee name is a number, a name exception must be thrown. b) If an employee age is greater than 50, an age exception must be thrown. c) Or else an object must be created for the entered employee details

Code:

```
import java.util.*;
class myexception extends Exception{
}
class employee{
    String name;
    int age;
    employee(String name,int age){
         this.name = name;
         this.age  =age;
    }
}
public class exp9a{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int flag=0;
            System.out.println("Enter your name ");
            String name = sc.next();
        try{
             char [] na= name.toCharArray();
             for(int i=0;i<na.length;i++){
                 if (Character.isDigit(na[i])){
                     flag+= 1;
                     throw  new myexception();
                 }
             }
        }
        catch (myexception  e){
            System.out.println("NameFormatException: Enter the name in correct
format!");
        }
        System.out.println("Enter your age ");
        int n = sc.nextInt();
        try{
            if (n>50){
```

```
                flag +=1;
                throw new myexception();
            }
        }
        catch(myexception d){
            System.out.println("AgeExceededException: You have exceeded the
age");
        }
        // employee e = new employee(name,n);
        // System.out.println(e.name);
        if (flag==0){
            employee e = new employee(name,n);
            System.out.println("Name of employee: "+e.name);
            System.out.println("Age of employee : "+e.age);
        }
        else{
            System.out.println("Please put a proper input!");
        }

    }
}
```

Output:

```
Enter your name
Adwait
Enter your age
56
AgeExceededException: You have exceeded the age
Please put a proper input!
```

```
Enter your name
Rohit
Enter your age
34
Name of employee: Rohit
Age of employee : 34
```

Q2:

There is a abstract class Account

Attribute:-

- **Name**
- **Balance**
- **AccNo**

Method:-

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

Checking Account and Saving Account inherits the Account class and provides the implementation for the methods accordingly

Saving Account class

Attribute:-

- **interestRate**
- **minBalance**
- **month**
- **FD_Balance**

Method

- **addInterest: handle Arithmetic Exception**
- **transfer():**

Checking Account class

Attribute:-

- **freeTransaction - number of free transactions**
- **transactionCost**
- **transactionCount - after every withdraw, deposit or transfer count is incremented(initially 0)**

Method

- **deductFee - if number of transaction greater than free transactions then transaction cost is deducted from the balance**

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.**
- **You can set the values by default for the above variables in Checking Account class**
- **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 failures are negative-amount-exception (in both deposit and withdraw transaction) and insufficient-amount-exception (in withdraw transaction).**
- **you may deposit money as FD or normal balance. If the amount is FD the call add interest method to add interest on the final amount**
- **withdraw will withdraw amount from Normal balance only**

For the above scenario write an interactive program in Java. Also, show output for different use cases

Code:

```
import java.util.Scanner;
class negativeAmount extends Exception{}
abstract class Account{
    String name;
    float Balance;
    int Accno=123456;
    int flag=1;

    abstract void Deposit();
    abstract void withdraw();
    abstract void display();
}
class Saving_Account extends Account{
    float interestRate=2, minBalance=0,FdBal;
    int month;

    void addInterest(){
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter the amount to be deposited in FD: ");
        float c=sc.nextFloat();
        try{
            if(c>=0){
                FdBal+=c;
                FdBal+=(FdBal * interestRate / 100);
                System.out.println("Your transaction was succesfull");
            }
            else{
                throw new negativeAmount();
            }
        }
        catch(negativeAmount na){
            System.out.println("Your transaction failed because of negative amount
exception");
```

```
        }
    }

    void transfer(){
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter the amount to be transferred: ");
        float d=sc.nextFloat();
        try{
            if(d>=0){
                Balance-=d;
                System.out.println("Your transaction was succesfull");
            }
            else{
                throw new negativeAmount();
            }
        }
        catch(negativeAmount na){
            System.out.println("Your transaction failed because of negative amount
exception");
        }
    }
    @Override
    void Deposit() {
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter the amount to be deposited: ");
        float e=sc.nextFloat();
        try{
            if(e>=0){
                Balance+=e;
                System.out.println("Your transaction was succesfull");
            }
            else{
                throw new negativeAmount();
            }
        }
        catch(negativeAmount na){
            System.out.println("Your transaction failed because of negative amount
exception");
        }
    }

    @Override
    void withdraw() {
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter the amount to be withdrawn: ");
        float f=sc.nextFloat();
        try{
            if(f>=0){
                Balance-=f;
                System.out.println("Your transaction was succesfull");
            }
            else{
                throw new negativeAmount();
            }
```

```
        }
        catch(negativeAmount na){
            System.out.println("Your transaction failed because of negative amount
exception");
        }
    }

    @Override
    void display() {
        System.out.println("Name of account holder: "+name);
        System.out.println("Account no.: "+Accno);
        System.out.println("Balance remaining in account: "+Balance);
        System.out.println("Balance in FD: "+FdBal);
    }

}
class Checking_Account extends Account{
    int freeTransaction=3, transactionCount=0;
    float transactionCost=10,a,b;

    void deductfee(){
        if(transactionCount>=freeTransaction){
            System.out.println("Transaction count exceeded 3, so fee for transaction
will be deducted.");
            try{
                if(Balance-transactionCost<0){
                    throw new negativeAmount();
                }
                else{
                    Balance-=transactionCost;
                }
            }
            catch(negativeAmount np){
                flag=0;
            }
        }
    }
    @Override
    void Deposit() {
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter the amount to be deposited: ");
        a=sc.nextFloat();
        try{
            if(a>=0){
                deductfee();
                if(flag==0){
                    System.out.println("Your transaction failed");
                    return;
                }
                Balance+=a;
                transactionCount++;
                System.out.println("Your transaction was succesfull");
            }
            else{
```

```
                throw new negativeAmount();
            }
        }
        catch(negativeAmount na){
            System.out.println("Your transaction failed because of negative amount
exception");
        }
    }

    @Override
    void withdraw() {
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter the amount to be withdrawn: ");
        b=sc.nextFloat();
        try{
            if(b>=0){
                deductfee();
                if(flag==0){
                    System.out.println("Transaction unsuccesfull");
                    return;
                }
                Balance-=b;
                transactionCount++;
                System.out.println("Your transaction was succesfull");
            }
            else{
                throw new negativeAmount();
            }
        }
        catch(negativeAmount na){
            System.out.println("Your transaction failed because of negative amount
exception");
        }
    }

    @Override
    void display() {
        System.out.println("Name of account holder: "+name);
        System.out.println("Account no.: "+Accno);
        System.out.println("Balance remaining in account: "+Balance);
    }

}
public class exp9a {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        Saving_Account sa=new Saving_Account();
        Checking_Account ca=new Checking_Account();
        System.out.println("Welcome to Axis Bank!");
        System.out.print("Enter the type of account: \n1) Savings Account\n2) Account
Check\n");
        int a=sc.nextInt();
        int b=0;
        if(a==1){
```

```
            System.out.print("Enter the name of the Account holder: ");
            sa.name=sc.next();
            while(b!=6){
                System.out.println("\n1. Deposit money\n2. Withdraw money\n3. Transfer
money to another linked account\n4. Deposit money in FD\n5. Display info\n6. Exit\n");
                b=sc.nextInt();
                if(b==1){
                    sa.Deposit();
                }
                else if(b==2){
                    sa.withdraw();
                }
                else if(b==3){
                    sa.transfer();
                }
                else if(b==4){
                    sa.addInterest();
                }
                else if(b==5){
                    sa.display();
                }
                else return;
            }
        }
        else{
            System.out.print("Enter the name of the Account holder: ");
            ca.name=sc.next();
            while(b!=6){
                System.out.println("\n1. Deposit money\n2. Withdraw money\n3. Display
info\n4. Exit\n");
                b=sc.nextInt();
                if(b==1){
                    ca.Deposit();
                }
                else if(b==2){
                    ca.withdraw();
                }
                else if(b==3){
                    ca.display();
                }
                else return;
            }
        }
    }
}
```

Output:

```
Welcome to Axis Bank!
Enter the type of account:
1) Savings Account
2) Account Check
1
Enter the name of the Account holder: Adwait

1. Deposit money
2. Withdraw money
3. Transfer money to another linked account
4. Deposit money in FD
5. Display info
6. Exit

1
Enter the amount to be deposited: 2345
Your transaction was succesfull
```

```
2
Enter the amount to be withdrawn: 1000
Your transaction was succesfull

1. Deposit money
2. Withdraw money
3. Transfer money to another linked account
4. Deposit money in FD
5. Display info
6. Exit

6
PS C:\Users\aspur\OneDrive\C PROGRAMS\mydirectory>
```