OOPS PRACTICAL 9

Name: Devasy Patel

Roll No: 20BCE057

Course Code: 2CS302

Course Name: Object Oriented Programming

Practical 9A

```
package p1;
//oops 9a
interface Polygon {
    void calcArea();
    void calcPerimeter();
    void display();
}
 public class Square implements Polygon{
    float s,a,p;
    public Square(float side){
        this.s=side;
    }
    @Override
    public void calcArea() {
        a=s*s;
    }
    @Override
   public void calcPerimeter() {
        p=4*s;
    }
    @Override
    public void display() {
        System.out.println("The Area of Square with side "+s+" is "+a);
        System.out.println("The Perimeter of Square with side "+s+" is "+p);
    }
}


//Different Class in same package

package p1;

public class Rectangle implements Polygon{
    float len,bre,a,p;
   public Rectangle(float len,float bre){
        this.len=len;
        this.bre=bre;
    }
    @Override
    public void calcArea() {
        a=len*bre;
    }
    @Override
    public void calcPerimeter() {
        p=(2*len)+(2*bre);
    }
    @Override
    public void display() {
        System.out.println("The Area of Rectangle with length "+len+" and breadth
"+bre+" is "+a);
```

```
        System.out.println("The Perimeter of Rectangle with length "+len+" and
breadth "+bre+" is "+p);
    }
}

//Different Package and importing package p1

package thread;
import p1.*;
import java.util.*;
public class oops_9a {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        System.out.print("enter side length for square:");
        Square sq = new Square(sc.nextFloat());
        sq.calcArea();
        sq.calcPerimeter();
        sq.display();
        System.out.print("enter length and breadth for rectangle:");
        Rectangle rect = new Rectangle(sc.nextFloat(),sc.nextFloat());
        rect.calcArea();
        rect.calcPerimeter();
        rect.display();
    }
}
```

OUTPUT

```
Run: oops_9a (1)
"C:\Program Files\Java\jdk-16.0.1\bin\java.exe" "-javaagent:C:\Program Files\Jet
enter side length for square:32
The Area of Square with side 32.0 is 1024.0
The Perimeter of Square with side 32.0 is 128.0
enter length and breadth for rectangle:32 32
The Area of Rectangle with length 32.0 and breadth 32.0 is 1024.0
The Perimeter of Rectangle with length 32.0 and breadth 32.0 is 128.0

Process finished with exit code 0
```

THEORETICAL PRINCIPLES USED:

In this practical we learnt concept of Interface and importing packages.

Practical 9B

```java
package p1;
//oops 9b
import java.util.*;
class CalAverage{
    double sum=0,avg;
    public double avgFirstN(int n){
        for(int i=1;i<=n;i++){
            sum+=i;
        }
        avg=sum/n;
        return avg;
    }
}
public class oops_9b {
    public static void main(String[] args) throws IllegalAccessException {
        boolean ch=true;
        Scanner sc = new Scanner(System.in);
        while (ch) {
            System.out.println("Enter your Number:");
            int num;
            try {
                num = sc.nextInt();
                if (num <= 0) {
                    throw new IllegalArgumentException("Number is not Natural
Number!!");
                }
                CalAverage c = new CalAverage();
                double ans = c.avgFirstN(num);
                System.out.println("Average :" + ans);
            } catch (IllegalArgumentException e) {
                System.out.println(e);
            }
            System.out.print("Do you want to continue:(1/0)");
            int c=sc.nextInt();
            if(c==1){
                ch=true;
            }
            else if(c==0){
                ch=false;
            }
        }
    }
}
```

OUTPUT

```
Run:  oops_9b ×
"C:\Program Files\Java\jdk-16.0.1\bin\java.exe" "-javaagent:C:\Prog
Enter your Number:
-4
java.lang.IllegalArgumentException: Number is not Natural Number!!
Do you want to continue:(1/0)1
Enter your Number:
5
Average :3.0
Do you want to continue:(1/0)0

Process finished with exit code 0
```

THEORETICAL PRINCIPLES USED:

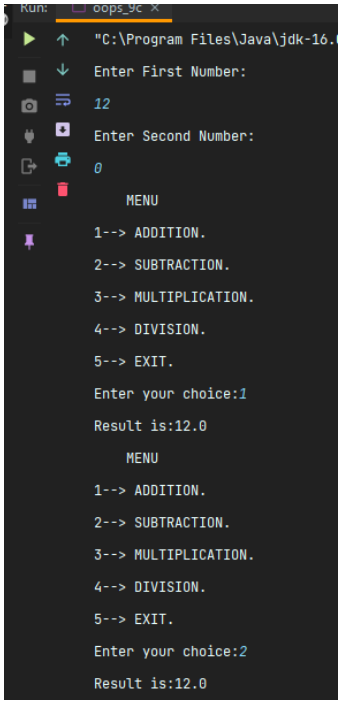
In this practical we learnt exception handling using keywords like try ,catch and throw.

Practical 9C

```
package p1;
import java.util.*;
class Number {
    int FirstNum, SecondNum;
    double result;

    Number(int x, int y) {
        this.FirstNum = x;
        this.SecondNum = y;
    }

    void add() {
        result = this.FirstNum + this.SecondNum;
    }

    void sub() {
        result = this.FirstNum - this.SecondNum;
    }

    void mul() {
        result = this.FirstNum * (this.SecondNum);
    }

    void div() {
        int flag = 0;
        try {
            if (this.SecondNum == 0) {
                flag = 1;
                throw new Exception("Divison by 0 is not allowed");
            }
        } catch (Exception e) {

        }
        if (flag == 1) {
            System.exit(0);
        } else {
            result = this.FirstNum / this.SecondNum;
        }
    }
}

    public class oops_9c {
        public static void main(String[] args) {
            Scanner sc = new Scanner(System.in);
            boolean a1 = true;
            System.out.println("Enter First Number:");
            int a = sc.nextInt();
            System.out.println("Enter Second Number:");
            int b = sc.nextInt();
            Number n = new Number(a, b);
            while (a1) {
                System.out.println("\tMENU");
                System.out.println("1--> ADDITION.");
                System.out.println("2--> SUBTRACTION.");
                System.out.println("3--> MULTIPLICATION.");
                System.out.println("4--> DIVISION.");
                System.out.println("5--> EXIT.");
```

```java
                System.out.print("Enter your choice:");
                int ch = sc.nextInt();
                switch (ch) {
                    case 1:
                        n.add();
                        break;
                    case 2:
                        n.sub();
                        break;
                    case 3:
                        n.mul();
                        break;
                    case 4:
                        n.div();
                        break;
                    case 5:
                        System.exit(0);
                }
                System.out.println("Result is:" + n.result);
            }
        }
    }
```

OUTPUT



THEORETICAL PRINCIPLES USED:

In this practical we learnt exception handling using keywords like try ,catch and throw.

Practical 9D

```
package p1;
import java.util.*;
class BankAccount{
    int accNo;
    String custName,accType;//Saving s,Current c
    float balance=0.0f;
    BankAccount(int an,String ct,String at,float bal){
        this.accNo=an;
        this.custName=ct;
        this.accType=at;
        this.balance=bal;
    }
    void deposit(float amt) {
        try {
        if (amt < 0) {
            throw new NegativeAmount("Amount is Negative!!");
        }
        else{
            this.balance+=amt;
        }
            } catch (NegativeAmount n) {
                System.out.println(n);
            }
    }
    void withdraw(float amt) {
        try {
            if (this.accType.equals("saving")) {//For savings account
                if ((this.balance - amt) > 1000) {
                    this.balance = getBalance() - amt;
                } else {
                    throw new InsufficientFunds("Insufficient Funds in your
Account!!");
                }
            }
        } catch (InsufficientFunds i) {
            System.out.println(i);
        }
        try {
            if (this.accType.equals("current")) {
                if ((this.balance - amt) > 5000) { // For Current Account
                    this.balance = getBalance() - amt;
                } else {
                    throw new InsufficientFunds("Insufficient Funds in your
Account!!");
                }
            }
        } catch (InsufficientFunds i) {
            System.out.println(i);
        }
    }
    float getBalance(){
        try {
            if (this.accType.equals("saving")) {//For savings account
                if (this.balance < 1000) {

                    throw new LowBalanceException("Low Balance in your Account!!");
                }
            }
        }
        catch(LowBalanceException l){
                    System.out.println(l);
                }
        try {
            if (this.accType.equals("current")) {//For Current account
                if (this.balance < 5000) {
```

```
                    throw new LowBalanceException("Low Balance in your Account!!");
                }
            }
        }catch(LowBalanceException l){
                    System.out.println(l);
                }
            return balance;
        }

    class NegativeAmount extends Exception{
        NegativeAmount(String s){
            super(s);
        }
    }
    class InsufficientFunds extends Exception{
        InsufficientFunds(String i){
            super(i);
        }
    }
    class LowBalanceException extends Exception{
        LowBalanceException(String l){
            super(l);
        }
    }
}
public class oops_9d {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Scanner sc1 = new Scanner(System.in);
        System.out.print("Enter your Account No: ");
        int ac = sc.nextInt();
        System.out.println("Enter your Name:");
        String nam = sc.next();
        System.out.print("Enter your Balance:");
        float bal = sc.nextFloat();
        System.out.println("Enter your Account Type:");
        String at = sc1.next();
        BankAccount ba = new BankAccount(ac, nam, at, bal);
        while (true) {
            System.out.println("1--> Deposit.\n2--> Withdraw.\n3--> Get
Balance.\n4--> Exit.");
            System.out.print("Enter your choice: ");
            int ch = sc.nextInt();
            switch (ch) {
                case 1:
                    System.out.println("Enter Amount for deposit:");
                    ba.deposit(sc.nextInt());
                    break;
                case 2:
                    System.out.print("Enter amount for withdrawal: ");
                    ba.withdraw(sc.nextInt());
                    break;
                case 3:
                   float b=ba.getBalance();
                    System.out.println(b);
                    break;
                case 4:
                    System.exit(0);
            }
        }
    }
}
```

OUTPUT

```
1--> Deposit.
2--> Withdraw.
3--> Get Balance.
4--> Exit.
Enter your choice: 2
Enter amount for withdrawal: 1500
p1.BankAccount$InsufficientFunds: Insufficient Funds in your Account!!
1--> Deposit.
2--> Withdraw.
3--> Get Balance.
4--> Exit.
Enter your choice: 3
2000.0
1--> Deposit.
2--> Withdraw.
3--> Get Balance.
4--> Exit.
Enter your choice: 4

Process finished with exit code 0
```

THEORETICAL PRINCIPLES USED:

In this practical we use custom exception and exception handling using keywords like try ,catch and throw.

Practical 9E

```
package p1;
import java.util.*;
class Emp{
    int empid;
    String empName,designation;
    double basic,HRA;
    Emp(int eid,String en,String des,double bas) throws LowSalException{
        this.empid=eid;
        this.empName=en;
        this.designation=des;
        this.basic=bas;
        if(this.basic<500){
            throw new LowSalException();
        }
    }
    void printDEL(){
        System.out.println("Name: "+this.empName);
        System.out.println("ID: "+this.empid);
        System.out.println("Designation: "+this.designation);
        System.out.println("Basic Salary: "+this.basic);
        System.out.println("HRA: "+this.HRA);
    }
    void calculateHRA(){
        if(this.designation.equals("Manager")){
            this.HRA=(0.1*this.basic);
        }
        if(this.designation.equals("Officer")){
            this.HRA=(0.12*this.basic);
        }
        if(this.designation.equals("CLERK")){
            this.HRA=(0.05*this.basic);
        }
    }
}
public class oops_9e {
    public static void main(String[] args) {
        boolean ch;
        do {
        Scanner sc = new Scanner(System.in);
        Scanner sc1 = new Scanner(System.in);
        System.out.print("Enter Id: ");
        int eid = sc.nextInt();
        System.out.print("Enter Name: ");
        String en = sc1.nextLine();
        System.out.print("Enter Designation: ");
        String des = sc1.nextLine();
        System.out.print("Enter Basic: ");
        double bas = sc.nextFloat();

            try {
                Emp e = new Emp(eid, en, des, bas);
                e.calculateHRA();
                e.printDEL();
            } catch (LowSalException e) {
                System.out.println(e);
            }
             System.out.println("Enter your choice(T/F):");
            ch=sc.nextBoolean();
        }while(ch);
    }
}
class LowSalException extends Exception{
    public String toString(){
        return "Basic is less than 500..";
```

```
    }
}
```

OUTPUT

```
Run: oops_9e
"C:\Program Files\Java\jdk-16.0.1\bi
Enter Id: 101
Enter Name: Manish Pandey
Enter Designation: Officer
Enter Basic: 499
Basic is less than 500..
Enter your choice(T/F):
true
Enter Id: 020
Enter Name: Aryan Pandi
Enter Designation: Manager
Enter Basic: 600
Name: Aryan Pandi
ID: 20
Designation: Manager
Basic Salary: 600.0
HRA: 60.0
Enter your choice(T/F):
false

Process finished with exit code 0
```

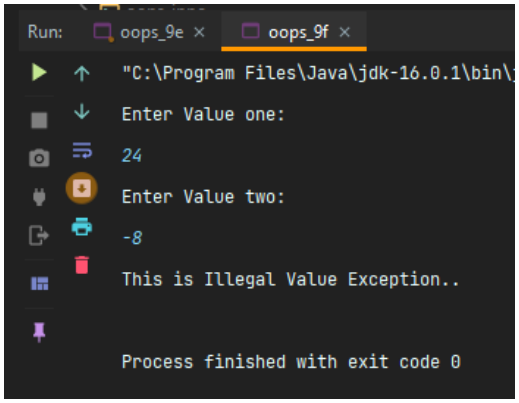
THEORETICAL PRINCIPLES USED:

In this practical we use custom exception and exception handling using keywords like try ,catch and throw.

Practical 9F

```java
package p1;
import java.util.*;
class USERTRAIL{
    int val1,val2;
    USERTRAIL(int v1,int v2) throws IllegalValueException{
        this.val1=v1;
        this.val2=v2;
        if(val1<0){
            throw new IllegalValueException();
        }
        if(val2<0){
            throw new IllegalValueException();
        }
    }
    public boolean show(){
        if((this.val1>0) && (this.val2>0)){
            return true;
        }
        else{
           return false;
        }
    }
}
class IllegalValueException extends Exception{
    public String toString(){
        return "This is Illegal Value Exception..";
    }
}
public class oops_9f {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter Value one:");
        int v1=sc.nextInt();
        System.out.println("Enter Value two:");
        int v2=sc.nextInt();
        try {
            USERTRAIL ut = new USERTRAIL(v1, v2);
            boolean bool=ut.show();
            if(bool){
                System.out.println("Both are greater than 0.");
            }
            else{
                System.out.println("None are greater than 0.");
            }
        }catch(IllegalValueException i){
            System.out.println(i);
        }
    }
}
```

OUTPUT



THEORETICAL PRINCIPLES USED:

In this practical we use custom exception and exception handling using keywords like try ,catch and throw.