**Lab Session 08**

**Apply the concepts of Interfaces**

**Date of the Session: Time of the Session:**

**Pre-Lab Tasks:**

**1. What is an interface?**

**2. Give an example of an interface definition.**

**3. How will you implement an interface?**

**4. Give an example for calling the method through interface reference variable.**

**5. What is nested interface?**

**In-Lab Tasks:**

1. Create an interface “Number” with the following abstract methods isZero( ), isPositive(), isNegative( ), isOdd( ), isEven( ), isPrime(), isAmstrong() the above methods return boolean primitive type. Implement this interface in “Verification” class.

**Test Case:**

***Test Case 1:***

***Input:***

151

***Output:***

positive and prime

***Test Case 2:***

***Input:***

-153

***Output:***

negative and Armstrong

**Aim:**

To create a interface that checks whether given number is Zero, positive, negative, odd, even, prime.

**Algorithm:**

1.Start.

2.Create a interface with name as number that contains methods with return type Boolean.

3.Create a class with name verification that implements interface number.

4.Inside the class, create a method that finds whether given number is zero, positive, negative.

5.Also create methods that checks whether given number is odd, even, prime, Armstrong.

6.Create a class with name NumberType that uses these methods to finds the given number type.

7.Create an object for Verification class and call those methods using the object.

8.Display the respective outputs.

9.Stop.

**Source Code:**

**NumberType.java**

import java.util.\*;

interface Number

{

boolean isZero(int n);

boolean isNegative(int n);

boolean isPositive(int n);

boolean isOdd(int n);

boolean isEven(int n);

boolean isPrime(int n);

boolean isArmstrong(int n);

}

class Verification implements Number

{

public boolean isZero(int n)

{

return n==0;

}

public boolean isNegative(int n)

{

return n<0;

}

public boolean isPositive(int n)

{

return n>0;

}

public boolean isOdd(int n)

{

return n%2!=0;

}

public boolean isEven(int n)

{

return n%2==0;

}

public boolean isPrime(int n)

{

int flag=0;

if(n==0||n==1)

{

return false;

}

else

{

for(int i=2;i<=n/2;i++)

{

if(n%i==0)

{

flag=1;

break;

}

}

if(flag==0)

{

return true;

}

return false;

}

}

public boolean isArmstrong(int n)

{

int r,sum=0,temp=n;

String s=Integer.toString(n);

int l=s.length();

while (n!= 0)

{

r = n % 10;

sum += Math.pow(r,l);

n /= 10;

}

return temp==sum;

}

}

class NumberType

{

public static void main(String[] args)

{

Verification obj=new Verification();

Scanner sc=new Scanner(System.in);

System.out.print("Enter the number ");

int n=sc.nextInt();

if(obj.isZero(n))

{

System.out.print("Zero ");

}

if(obj.isPositive(n))

{

System.out.print("Positive ");

}

if(obj.isNegative(n))

{

System.out.print("Negative ");

}

if(obj.isOdd(n))

{

System.out.print("Odd ");

}

if(obj.isEven(n))

{

System.out.print("Even ");

}

if(obj.isPrime(n) && obj.isPositive(n))

{

System.out.print("Prime ");

}

if(obj.isArmstrong(n) && obj.isPositive(n))

{

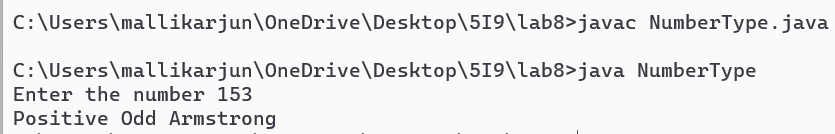
System.out.print("Armstrong ");

}

}

}

**Output:**

****

**Result:**

The program to find what is the type of given number is successfully executed using interface.

1. Write a program to create interface named test. In this interface the member function is square. Implement this interface in arithmetic class. Create one new class called ToTestInt in this class use the object of arithmetic class.

**Test Cases:**

***Test Case 1:***

***Input:***

1424

***Output:***

2027776

***Test Case 2:***

***Input:***

-12345

***Output:***

152399025

**Aim:**

To implement an interface that finds the square of given number.

**Algorithm:**

1.Start

2.Create a interface named as test.

3.Inside the interface, create a abstract method with name square that returns square of given number.

4.Create a class arithmetic that implements the interface test.

5.Create another class ToTestInt that tests the interface.

6.Create object for arithmetic class and prompt the user to enter a number.

7.Call the method and display the respective output.

8.Stop.

**Source Code:**

**ToTestInt.java**

import java.util.Scanner;

interface test

{

long square(long n);

}

class Arithmetic implements test

{

public long square(long n)

{

return n\*n;

}

}

class ToTestInt

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

Arithmetic obj=new Arithmetic();

System.out.print("Enter the number ");

long num=sc.nextLong();

System.out.println(obj.square(num));

}

}

**Output:**

****

**Result:**

The program to find square using interface is successfully executed.

**Post-Lab Tasks:**

1. Write an interface named “EmployeeDetails”which consist of following abstract methods 1. Enter Data 2. Display Data 3. Exit. Implement this interface in “Userselection” class to invoke the respective method according to the given menu input.

**Test Cases:**

***Test Case 1:***

***Input:***

{‘Varun’, 1245, 60000}

***Output:***

Employee Name= ‘Varun’

Employee Id= 1245

Employee Salary= 60000

***Test Case 2:***

***Input:***

{‘John. 1245’, 12345}

***Output:***

Invalid number of arguments

**Aim:**

To print the employee details using concept of interface and user input.

**Algorithm:**

1.Start.

2.Create an interface with name Employee.

3.Create a class with name UserSelection that implements the Employee interface.

4.Create two methods with names ReadData, DisplayData that reads and displays data.

5.Create a class with name EmployeeTest that uses the above class.

6.Create object for UserSelection class and call the methods using the created object.

7.Run an infinite loop.

8.If user enter 1, read the data.

9.If user enter 2,Display the data.

10.If user enter 3, terminate the program.

11.Stop.

**Source Code:**

**EmployeeTest.java**

import java.util.Scanner;

interface Employee

{

void ReadData();

void DisplayData();

}

class UserSelection implements Employee

{

Scanner sc=new Scanner(System.in);

String name;

int empid,salary;

public void ReadData()

{

System.out.print("Enter employee name ");

name=sc.next();

System.out.print("Enter employee id ");

empid=sc.nextInt();

System.out.print("Enter salary ");

salary=sc.nextInt();

System.out.println();

}

public void DisplayData()

{

System.out.println("Employee name="+name);

System.out.println("Employee id="+empid);

System.out.println("Employee salary="+salary);

System.out.println();

}

}

class EmployeeTest

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

UserSelection obj=new UserSelection();

int choice;

System.out.println("1.Read Data\n2.Display data\n3.Exit");

while(true)

{

System.out.print("Enter ur choice ");

choice=sc.nextInt();

switch (choice)

{

case 1:

obj.ReadData();

break;

case 2:

obj.DisplayData();

break;

case 3:

System.out.println("Exiting....");

System.exit(0);

break;

}

}

}

}

**Output:**

****

**Result:**

The program to display employee details using interface is executed successfully.

**Student’s Signature**

***(For Evaluator’s use only)***

**Marks Secured: \_\_\_\_\_\_\_\_\_ out of \_\_\_\_\_\_\_\_\_\_**

**Faculty Signature**