CSIT Department

Bsc. CSIT 6th Sem



**Bhaktapur Multiple Campus**

Dudhpati – 17, Bhaktapur

**Lab Assignment of Advanced Java**

**(CSC 409)**

**Submitted By Submitted To**

Saini Thapa Magar Lecturer,

Roll No: 23264/076 Karan Shrestha

**Lab Assignment 1**

1. An array is called balanced if its even numbered elements (a[0], a[2], etc.) are even and its odd numbered elements (a[1], a[3], etc.) are odd. Write a function named balanced that accepts an array of integers which returns 1 if the array is balanced and returns 0 otherwise. [2075]

**Solution**:

public class qn1{

public int balancedornot(int[] numbers)

{

for(int i=0;i<numbers.length;i++){

if(i%2==0 && numbers[i]%2==0){

continue;

}

else if(i%2==1 && numbers[i]%2==1){

continue;

}

else

return 0;

}

return 1;

}

public static void main(String args[]){

qn1 n = new qn1();

int[] numbers={3,4,5,6,7,8};

System.out.println(n.balancedornot(numbers));

}

}

1. Write an object oriented program to find area and perimeter of rectangle. [2073, 2074]

**Solution:**

import java.util.\*;

public class qn2{

public void perimeter(int a,int b){

System.out.println("Perimeter = " + (a+b));

}

public void area (int a, int b){

System.out.println("Area = " + (a\*b));

}

public static void main(String args[]){

qn2 a1= new qn2();

Scanner inp= new Scanner(System.in);

System.out.println("Enter the length and breadth of the rectangle: \n");

int a=inp.nextInt();

int b=inp.nextInt();

a1.perimeter(a,b);

a1.area(a,b);

}

}

1. Write a program to input and add two numbers using static methods (procedural programming).

**Solution:**

import java.util.Scanner;

public class qn3{

public static int addNum(int a, int b){

return a+b;

}

public static void main(String args[]){

Scanner scan= new Scanner(System.in);

System.out.println("Enter the value of a and b: ");

int a = scan.nextInt();

int b = scan.nextInt();

System.out.println("The sum of two numbers is " + qn3.addNum(a,b));

}

}

1. Write a program to input principle, time and rate, then calculate simple interest using static methods.

**Solution:**

import java.util.Scanner;

public class qn4{

public static float SimpleInterest(float rate, int time, int principal){

return (rate\*time\*principal)/100;

}

public static void main(String args[]){

Scanner SI=new Scanner(System.in);

System.out.println("Enter Principal amount: ");

int principal = SI.nextInt();

System.out.println("Enter Interest rate: ");

float rate = SI.nextFloat();

System.out.println("Enter Time(in years): ");

int time = SI.nextInt();

System.out.println("Simple interest = "+ qn4.SimpleInterest(rate, time, principal));

}

}

1. Write both procedural and object oriented programs to calculate the area of a

a) Circle

* 1. Square
  2. Rectangle
  3. Sphere

**Solution:**

**OOP.java:**

package qn5;

import java.util.Scanner;

public class OOP{

public float Circle(int radius){

return 22\*radius\*radius/7;

}

public int rectangle(int a, int b){

return (a\*b);

}

public int Square(int a){

return a\*a;

}

public float sphere(int radius){

return 4\*22\*radius\*radius/7;

}

public static void main(String args[]){

OOP obj = new OOP();

Scanner s = new Scanner(System.in);

//Circle

System.out.println("Enter the radius of circle: ");

int radius = s.nextInt();

System.out.println("The area of circle : " + obj.Circle(radius));

// Rectangle

System.out.println("Enter the length and breadth of rectangle: ");

int length = s.nextInt();

int breadth = s.nextInt();

System.out.println("The area of rectangle : " + obj.rectangle(length,breadth));

// Square

System.out.println("Enter the length of square: ");

int length1 = s.nextInt();

System.out.println("The area of square : " + obj.Square(length1));

//Sphere

System.out.println("Enter the radius of sphere: ");

int radius1 = s.nextInt();

System.out.println("The area of sphere : " + obj.sphere(radius1));

}

}

**Procedural.java:**

package qn5;

import java.util.Scanner;

public class Procedural{

public static float Circle(int radius){

return 22\*radius\*radius/7;

}

public static int rectangle(int a, int b){

return (a\*b);

}

public static int Square(int a){

return a\*a;

}

public static float sphere(int radius){

return 4\*22\*radius\*radius/7;

}

public static void main(String args[]){

Scanner s = new Scanner(System.in);

//Circle

System.out.println("Enter the radius of circle: ");

int radius = s.nextInt();

System.out.println("The area of circle : " + Procedural.Circle(radius));

// Rectangle

System.out.println("Enter the length and breadth of rectangle: ");

int length = s.nextInt();

int breadth = s.nextInt();

System.out.println("The area of rectangle : " + Procedural.rectangle(length,breadth));

// Square

System.out.println("Enter the length of square: ");

int length1 = s.nextInt();

System.out.println("The area of square : " + Procedural.Square(length1));

//Sphere

System.out.println("Enter the radius of sphere: ");

int radius1 = s.nextInt();

System.out.println("The area of sphere : " + Procedural.sphere(radius1));

}

}

1. Write a static method to calculate the sum of a one dimensional array

**Solution:**

public class qn6 {

public static int AddArray(int[] numbers){

int sum=0;

for(int i=0;i<numbers.length;i++){

sum+=numbers[i];

}

return sum;

}

public static void main(String args[])

{

int[] numbers={2,3,5,6,8};

System.out.println("Sum of the array = "+qn6.AddArray(numbers));

}

}

1. Write a program to demonstrate encapsulation.

**Solution:**

//Encapsulation

public class qn7{

private int age;

private String name;

//constructor

public qn7(int age, String name) {

this.age = age;

this.name = name;

}

public String getName() {

return this.name;

}

public int getAge() {

return this.age;

}

public static void main(String[] args) {

qn7 e = new qn7(22,"Prisan");

int resultAge = e.getAge();

String name = e.getName();

System.out.println(resultAge);

System.out.println(name);

}

}

1. Write a program to demonstrate inheritance.

**Solution:**

**Baseclass,java**

package qn8;

public class Baseclass {

public void Tiger(){

System.out.println("ROAR");

}

}

**Derivedclass.java**

package qn8;

public class Derivedclass extends Baseclass{

public void DOG(){

System.out.println("WOOF");

}

public static void main(String args[]){

Derivedclass a1= new Derivedclass();

a1.DOG();

a1.Tiger();

}

}

1. Write a program to demonstrate polymorphism using interface as parent.

**Solution:**

**sounds.java**

package qn9;

interface sounds{

void Sound();

}

**Polymorphism.java**

package qn9;

public class Polymorphism implements sounds{

@Override

public void Sound(){

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

}

public static void main(String args[]){

Polymorphism p1= new Polymorphism();

p1.Sound();

}

}

1. Write a program to create two classes Circle and Square, with appropriate fields and methods, in a package name shape. Create a separate class ShapeDemo to test the classes.

**Solution:**

**Circle.java**

package shape;

public class circle {

private int radius;

public circle(int radius){

this.radius=radius;

}

double GetArea(){

return 22\*radius\*radius/7;

} }

**square.java**

package shape;

public class square {

private int length;

public square(int radius){

this.length=radius;

}

double GetArea(){

return length \* length;

}

}

**ShapeDemo.java**

package shape;

public class ShapeDemo {

public static void main(String args[]){

// ShapeDemo S1 = new ShapeDemo();

circle c= new circle(5);

System.out.println(c.GetArea());

square s= new square(10);

System.out.println(s.GetArea());

}

}

1. Write a program to demonstrate try-catch-finally.

**Solution:**

import java.util.Scanner;

public class qn11 {

public static void main(String args[]){

Scanner obj = new Scanner(System.in);

System.out.println("Enter the value of a : ");

int a = obj.nextInt();

System.out.println("Enter the value of b : ");

int b = obj.nextInt();

try{

int res = a/b;

String str = String.format("The quotient is %d",res);

System.out.println(str);

}

catch(Exception e){

System.out.println(e.getMessage());

}

finally{

int sum = a + b;

System.out.println("The Final result is " + sum);

}

}}

1. Write a program to create two threads. The first thread should print numbers from 1 to 10 at intervals of 0.5 second and the second thread should print numbers from 11 to 20 at the interval of 1 second.

**Solution:**

class NumberPrinter extends Thread{

private int start;

private int end;

private long miliseconds;

public NumberPrinter(int start,int end,long miliseconds){

this.start=start;

this.end=end;

this.miliseconds=miliseconds;

}

public void run() {

for (int i = start; i <= end; i++) {

System.out.print(i + " ");

try {

Thread.sleep(miliseconds);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

}

public class qn12{

public static void main(String args[]){

Thread p1= new NumberPrinter(1, 10, 500);

Thread p2 = new NumberPrinter(1, 10, 1000);

p1.start();

System.out.println("\n");

p2.start();

}

}

1. Write a program to execute multiple threads in priority base. [2075]

**Solution:**

class PThread extends Thread{

public PThread(String name){

super(name);}

public void run() {

for (int i = 1; i <= 5; i++) {

System.out.println(getName() + " is running, Count: "+i);

}

System.out.println("\n");

}

}

public class qn13 {

public static void main(String args[]){

PThread p1= new PThread("1st Thread");

PThread p2= new PThread("2nd Thread");

PThread p3= new PThread("3rd Thread");

//Set thread priority

p1.setPriority(Thread.MIN\_PRIORITY);

p2.setPriority(Thread.NORM\_PRIORITY);

p3.setPriority(Thread.MAX\_PRIORITY);

p1.start();

p2.start();

p3.start();

} }

1. Write the simple java program that reads data from one file and writes data to another file. [2070, 2071, 2073, 2074]

**Solution:**

import java.io.\*;

public class qn14 {

public static void main(String args[])

{

String sourcefile = "C:\\Users\\LENOVO\\Desktop\\7th sem\\Advanced Java\\Java program\\UNIT 1\\Saini.txt";

String DestFile = "C:\\Users\\LENOVO\\Desktop\\7th sem\\Advanced Java\\Java program\\UNIT 1\\Thapa.txt";

try (BufferedReader FR = new BufferedReader(new FileReader(sourcefile));

BufferedWriter FW = new BufferedWriter(new FileWriter(DestFile))){

String line;

while((line=FR.readLine())!=null){

FW.write(line);

}

System.out.println("File copied successfully");

}

catch(IOException e){

System.out.println("An error occured : " + e.getMessage());

}

}

}

1. Write a program to duplicate each character in a text file and write the output in a  
   separate file using character stream.

**Solution:**

import java.io.\*;

public class qn15 {

public static void main(String[] args) {

String sourcefile = "C:\\Users\\LENOVO\\Desktop\\7th sem\\Advanced Java\\Java program\\UNIT 1\\Saini.txt";

String DestFile = "C:\\Users\\LENOVO\\Desktop\\7th sem\\Advanced Java\\Java program\\UNIT 1\\Thapa.txt";

try (FileReader reader = new FileReader(sourcefile);

FileWriter writer = new FileWriter(DestFile)) {

int charRead;

while ((charRead = reader.read()) != -1) {

writer.write(charRead);

}

System.out.println("Characters duplicated successfully!");

} catch (IOException e) {

e.printStackTrace();

}

}}

1. Write a program to read records from a text file which contains people’s name,  
   principle, rate and time values. Calculate simple interest and write all the contents of the  
   source file along with simple interest to destination file

**Solution:**

import java.io.\*;

public class qn16{

public static void main(String args[]){

String sourcefile = "C:\\Users\\LENOVO\\Desktop\\7th sem\\Advanced Java\\Java program\\UNIT 1\\source.txt";

String DestFile = "C:\\Users\\LENOVO\\Desktop\\7th sem\\Advanced Java\\Java program\\UNIT 1\\dest.txt";

try (BufferedReader reader = new BufferedReader(new FileReader(sourcefile));

BufferedWriter writer = new BufferedWriter(new FileWriter(DestFile))) {

String line;

while ((line = reader.readLine()) != null) {

String[] parts = line.split(",");

if (parts.length == 4) {

String name = parts[0].trim();

double principle = Double.parseDouble(parts[1].trim());

double rate = Double.parseDouble(parts[2].trim());

double time = Double.parseDouble(parts[3].trim());

double simpleInterest = (principle \* rate \* time) / 100.0;

String outputLine = name + ", " + principle + ", " + rate + ", " + time + ", " + simpleInterest;

writer.write(outputLine);

writer.newLine(); // Add a newline character to separate records

}

}

System.out.println("Simple interest calculated and written to the destination file!");

}

catch (IOException | NumberFormatException e) {

e.printStackTrace();

}

}

}

1. Write a program to read the contents of a file one line at a time and output them to the screen.

**Solution:**

import java.io.\*;

public class qn17 extends Thread{

public static void main(String args[])

{

String sourcefile = "C:\\Users\\LENOVO\\Desktop\\7th sem\\Advanced Java\\Java program\\UNIT 1\\dest.txt";

try{

BufferedReader reader = new BufferedReader(new FileReader(sourcefile));

String line;

while((line=reader.readLine())!=null){

System.out.println(line);

}

}

catch(IOException e){

System.err.println("An error occurred: " + e.getMessage());

}

}

}

1. Write a program to input whole lines from the keyboard and write them to a file. Exit the program when the user types “quit”.

**Solution:**

import java.io.\*;

public class qn18 {

public static void main(String args[])

{

String source = "C:\\Users\\LENOVO\\Desktop\\7th sem\\Advanced Java\\Java program\\UNIT 1\\Saini.txt";

boolean BoolLogic=true;

try{

BufferedWriter bw = new BufferedWriter(new FileWriter(source));

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

while(BoolLogic){

String line = br.readLine();

if("quit".equalsIgnoreCase(line.trim())){

BoolLogic=false;

}

else{

bw.write(line);

bw.newLine();

}

}

System.out.println("File Writing completed");

}

catch(IOException e){

System.err.println("An error occurred: " + e.getMessage());

}

}

}