

**School of Science and Technology**

**B.Sc. in Computer Science and Engineering**

**Assignment: 01**

|  |  |
| --- | --- |
| **Submitted By** | **Submitted To** |
| Name: **Md. Mehedi Hasan**  Student ID: 19-0-52-801-039  Course Title: Object Oriented Programming  Course Code: CSE2137 | **Samrat Kumar Dey**  School of Science and Technology  Bangladesh Open University  Gazipur-1705  Signature: |
| **Date of Submission: 30 December 2023** | |

**QUESTION-01:** write a java program that works as a simple calculator. Use a Grid Layout to arrange Buttons for digits and for the +, -, \* and % operations. Add a text field to display the result.

**SOLVE:**

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class SimpleCalculator extends JFrame implements ActionListener {

private JTextField result;

private JButton[] digits;

private JButton[] operators;

private char op;

private double num1;

private double num2;

private boolean start;

public SimpleCalculator() {

setTitle("Simple Calculator");

setSize(300, 300);

setLocation(500, 200);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

JPanel resultPanel = new JPanel();

result = new JTextField(20);

result.setEditable(false);

resultPanel.add(result);

JPanel buttonPanel = new JPanel();

buttonPanel.setLayout(new GridLayout(4, 4));

digits = new JButton[10];

operators = new JButton[5];

for (int i = 0; i < 10; i++) {

digits[i] = new JButton(String.valueOf(i));

buttonPanel.add(digits[i]);

}

operators[0] = new JButton("+");

operators[1] = new JButton("-");

operators[2] = new JButton("\*");

operators[3] = new JButton("/");

operators[4] = new JButton("%");

for (int i = 0; i < 5; i++) {

buttonPanel.add(operators[i]);

}

add(resultPanel, BorderLayout.NORTH);

add(buttonPanel, BorderLayout.CENTER);

op = ' ';

num1 = 0.0;

num2 = 0.0;

start = true;

for (int i = 0; i < 10; i++) {

digits[i].addActionListener(this);

}

for (int i = 0; i < 5; i++) {

operators[i].addActionListener(this);

}

}

public void actionPerformed(ActionEvent e) {

Object source = e.getSource();

if (source instanceof JButton) {

JButton button = (JButton) source;

String text = button.getText();

if (Character.isDigit(text.charAt(0))) {

if (start) {

result.setText("");

start = false;

}

result.setText(result.getText() + text);

}

else {

op = text.charAt(0);

num1 = Double.parseDouble(result.getText());

result.setText("");

}

}

else {

num2 = Double.parseDouble(result.getText());

switch (op) {

case '+':

num1 = num1 + num2;

break;

case '-':

num1 = num1 - num2;

break;

case '\*':

num1 = num1 \* num2;

break;

case '/':

num1 = num1 / num2;

break;

case '%':

num1 = num1 % num2;

break;

}

result.setText(String.valueOf(num1));

start = true;

}

}

public static void main(String[] args) {

SimpleCalculator calc = new SimpleCalculator();

calc.setVisible(true);

}

}

A screenshot of a computer

Description automatically generated

**QUESTION-02:** write a java program to find prime numbers between 1 to n.

**SOLVE:**

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Enter the value of n: ");

int n = input.nextInt();

System.out.println("Prime numbers between 1 and " + n + " are:");

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

boolean isPrime = true;

for (int j = 2; j <= Math.sqrt(i); j++) {

if (i % j == 0) {

isPrime = false;

break;

}

}

if (isPrime) {

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

}

}

}

}

A screenshot of a computer

Description automatically generated

**QUESTION-03:** write a java program that prints all real solutions to the quadratic equation ax^2+bx+c=0. Read in a,b,c and use the quadratic formula.

**SOLVE:**

import java.util.Scanner;

public class QuadraticSolver {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.println("Enter the values of a, b, and c for the quadratic equation ax^2+bx+c=0");

System.out.print("a: ");

double a = input.nextDouble();

System.out.print("b: ");

double b = input.nextDouble();

System.out.print("c: ");

double c = input.nextDouble();

double discriminant = b \* b - 4 \* a \* c;

if (discriminant > 0) {

double x1 = (-b + Math.sqrt(discriminant)) / (2 \* a);

double x2 = (-b - Math.sqrt(discriminant)) / (2 \* a);

System.out.println("The equation has two real solutions: " + x1 + " and " + x2);

} else if (discriminant == 0) {

double x = -b / (2 \* a);

System.out.println("The equation has one real solution: " + x);

} else {

System.out.println("The equation has no real solutions.");

}

input.close();

}

}

A screenshot of a computer

Description automatically generated

**QUESTION-04:** Create a base class Fruit which has name, taste and sixe as its attributes. A method called eat() is created which describes the name of the fruit and its taste. Inherit the same in 2 other class Apple and Orange and override the eat() method to represent each fruit taste.

**SOLVE:**

class Fruit {

String name;

String taste;

String size;

public Fruit(String name, String taste, String size) {

this.name = name;

this.taste = taste;

this.size = size;

}

public void eat() {

System.out.println("This is a " + name + " fruit. It tastes " + taste + ".");

}

}

class Apple extends Fruit {

public Apple(String size) {

super("Apple", "sweet", size);

}

public void eat() {

System.out.println("You are eating an apple. It tastes sweet and crunchy.");

}

}

class Orange extends Fruit {

public Orange(String size) {

super("Orange", "citrusy", size);

}

public void eat() {

System.out.println("You are eating an orange. It tastes citrusy and juicy.");

}

}

public class FruitTest {

public static void main(String[] args) {

Apple apple = new Apple("Medium");

Orange orange = new Orange("Large");

apple.eat();

orange.eat();

}

}

A screenshot of a computer

Description automatically generated

**QUESTION-05:** Write a java program to illustrate the concept of class with method overloading.

**SOLVE:**

public class MethodOverloadingExample {

public int add(int a, int b) {

return a + b;

}

public int add(int a, int b, int c) {

return a + b + c;

}

public double add(double a, double b) {

return a + b;

}

public String concatenate(String str1, String str2) {

return str1 + str2;

}

public static void main(String[] args) {

MethodOverloadingExample example = new MethodOverloadingExample();

System.out.println("Sum of 5 and 10: " + example.add(5, 10));

System.out.println("Sum of 5, 10, and 15: " + example.add(5, 10, 15));

System.out.println("Sum of 2.5 and 3.5: " + example.add(2.5, 3.5));

System.out.println("Here is the output ");

}

}

A screenshot of a computer

Description automatically generated

**QUESTION-06:** Write a java program to create an abstract class named Shape that contains two integers and an empty method named print Area(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the given shape.

**SOLVE:**

abstract class Shape {

int dimension1;

int dimension2;

public Shape(int dimension1, int dimension2) {

this.dimension1 = dimension1;

this.dimension2 = dimension2;

}

public abstract void printArea();

}

class Rectangle extends Shape {

public Rectangle(int length, int width) {

super(length, width);

}

public void printArea() {

int area = dimension1 \* dimension2;

System.out.println("Area of Rectangle: " + area);

}

}

class Triangle extends Shape {

public Triangle(int base, int height) {

super(base, height);

}

public void printArea() {

double area = 0.5 \* dimension1 \* dimension2;

System.out.println("Area of Triangle: " + area);

}

}

class Circle extends Shape {

public Circle(int radius) {

super(radius, 0); // In the case of a circle, only one dimension (radius) is needed

}

public void printArea() {

double area = Math.PI \* dimension1 \* dimension1;

System.out.println("Area of Circle: " + area);

}

}

public class ShapeTest {

public static void main(String[] args) {

Rectangle rectangle = new Rectangle(5, 8);

Triangle triangle = new Triangle(4, 6);

Circle circle = new Circle(3);

rectangle.printArea();

triangle.printArea();

circle.printArea();

}

}

A screenshot of a computer

Description automatically generated

**QUESTION-07:** Develop a java application with Employee class with Emp\_name, Emp\_id, Address, Mail\_id, Mobile\_no as members. Inherit the classes, Programmer, Assistant Professor, Associate Professor and Professor from employee class. And Basic pay (BP) as the member of all the inherited classes with 97% of BP as Dearness allowance (DA), 10% of BP as House Rent Allowance (HRA), 12% of BP as Provident fund (PF), 0.1% of BP for staff club fund. Generate pay slips for the employees with their gross and net salary.

**SOLVE:**

import java.util.Scanner;

class Employee {

String empName;

int empId;

String address;

String mailId;

long mobileNo;

public Employee(String empName, int empId, String address, String mailId, long mobileNo) {

this.empName = empName;

this.empId = empId;

this.address = address;

this.mailId = mailId;

this.mobileNo = mobileNo;

}

public void generatePaySlip(double basicPay) {

double da = 0.97 \* basicPay;

double hra = 0.10 \* basicPay;

double pf = 0.12 \* basicPay;

double staffClubFund = 0.001 \* basicPay;

double grossSalary = basicPay + da + hra;

double netSalary = grossSalary - pf - staffClubFund;

System.out.println("Employee Name: " + empName);

System.out.println("Employee ID: " + empId);

System.out.println("Address: " + address);

System.out.println("Mail ID: " + mailId);

System.out.println("Mobile Number: " + mobileNo);

System.out.println("Basic Pay: $" + basicPay);

System.out.println("Dearness Allowance (DA): $" + da);

System.out.println("House Rent Allowance (HRA): $" + hra);

System.out.println("Provident Fund (PF): $" + pf);

System.out.println("Staff Club Fund: $" + staffClubFund);

System.out.println("Gross Salary: $" + grossSalary);

System.out.println("Net Salary: $" + netSalary);

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

}

}

class Programmer extends Employee {

double basicPay;

public Programmer(String empName, int empId, String address, String mailId, long mobileNo, double basicPay) {

super(empName, empId, address, mailId, mobileNo);

this.basicPay = basicPay;

}

}

class AssistantProfessor extends Employee {

double basicPay;

public AssistantProfessor(String empName, int empId, String address, String mailId, long mobileNo, double basicPay) {

super(empName, empId, address, mailId, mobileNo);

this.basicPay = basicPay;

}

}

class AssociateProfessor extends Employee {

double basicPay;

public AssociateProfessor(String empName, int empId, String address, String mailId, long mobileNo, double basicPay) {

super(empName, empId, address, mailId, mobileNo);

this.basicPay = basicPay;

}

}

class Professor extends Employee {

double basicPay;

public Professor(String empName, int empId, String address, String mailId, long mobileNo, double basicPay) {

super(empName, empId, address, mailId, mobileNo);

this.basicPay = basicPay;

}

}

public class PayrollSystem {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

Programmer programmer = new Programmer("Mehedi", 101, "USA", "mehedi2@.com", 01756543210L, 5000);

AssistantProfessor assistantProfessor = new AssistantProfessor("Jane Smith", 201, "456 Oak St", "jane.smith@example.com", 9876543211L, 6000);

AssociateProfessor associateProfessor = new AssociateProfessor("Bob Johnson", 301, "789 Pine St", "bob.johnson@example.com", 9876543212L, 7000);

Professor professor = new Professor("Alice Brown", 401, "101 Elm St", "alice.brown@example.com", 9876543213L, 8000);

programmer.generatePaySlip(programmer.basicPay);

assistantProfessor.generatePaySlip(assistantProfessor.basicPay);

associateProfessor.generatePaySlip(associateProfessor.basicPay);

professor.generatePaySlip(professor.basicPay);

scanner.close();

}

}

A screenshot of a computer

Description automatically generated

**QUESTION-08:** Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the result field when the divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatExecption. If Num2 were zero, the program would throw an Arithmetic exception Display the exception in a message dialog box .

**SOLVE:**

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class IntegerDivision extends Frame implements ActionListener {

private TextField num1, num2, result;

private Button divide;

public IntegerDivision() {

setTitle("Integer Division");

setSize(300, 200);

setLocation(500, 200);

addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {

System.exit(0);

}

});

Panel inputPanel = new Panel();

inputPanel.setLayout(new GridLayout(3, 2));

num1 = new TextField(10);

num2 = new TextField(10);

result = new TextField(10);

result.setEditable(false);

inputPanel.add(new Label("Num1: "));

inputPanel.add(num1);

inputPanel.add(new Label("Num2: "));

inputPanel.add(num2);

inputPanel.add(new Label("Result: "));

inputPanel.add(result);

Panel buttonPanel = new Panel();

buttonPanel.setLayout(new FlowLayout());

divide = new Button("Divide");

buttonPanel.add(divide);

add(inputPanel, BorderLayout.CENTER);

add(buttonPanel, BorderLayout.SOUTH);

divide.addActionListener(this);

}

public void actionPerformed(ActionEvent e) {

Object source = e.getSource();

if (source == divide) {

String snum1 = num1.getText();

String snum2 = num2.getText();

try {

int n1 = Integer.parseInt(snum1);

int n2 = Integer.parseInt(snum2);

if (n2 == 0) {

throw new ArithmeticException("Division by zero");

}

int res = n1 / n2;

result.setText(String.valueOf(res));

} catch (NumberFormatException nfe) {

JOptionPane.showMessageDialog(this, "Invalid input. Enter an integer.", "Error", JOptionPane.ERROR\_MESSAGE);

} catch (ArithmeticException ae) {

JOptionPane.showMessageDialog(this, ae.getMessage(), "Error", JOptionPane.ERROR\_MESSAGE);

}

}

}

public static void main(String[] args) {

IntegerDivision id = new IntegerDivision();

id.setVisible(true);

}

}

A screenshot of a computer

Description automatically generated

**QUESTION-09:** write a java program that handles all mouse events and show the event name at the center of the window when a mouse event is fired.(use adapter classes).

**SOLVE:**

import java.awt.\*;

import java.awt.event.\*;

public class MouseEventsDemo extends Frame {

private Label label;

public MouseEventsDemo() {

setTitle("Mouse Events Demo");

setSize(300, 200);

setLocation(500, 200);

addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {

System.exit(0);

}

});

label = new Label("No mouse event");

label.setAlignment(Label.CENTER);

label.setFont(new Font("Arial", Font.BOLD, 24));

add(label, BorderLayout.CENTER);

addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e) {

label.setText("Mouse clicked at (" + e.getX() + ", " + e.getY() + ")");

}

public void mouseEntered(MouseEvent e) {

label.setText("Mouse entered at (" + e.getX() + ", " + e.getY() + ")");

}

public void mouseExited(MouseEvent e) {

label.setText("Mouse exited at (" + e.getX() + ", " + e.getY() + ")");

}

public void mousePressed(MouseEvent e) {

label.setText("Mouse pressed at (" + e.getX() + ", " + e.getY() + ")");

}

public void mouseReleased(MouseEvent e) {

label.setText("Mouse released at (" + e.getX() + ", " + e.getY() + ")");

}

});

}

public static void main(String[] args) {

MouseEventsDemo med = new MouseEventsDemo();

med.setVisible(true);

}

}

A screenshot of a computer

Description automatically generated

**QUESTION-10:** Develop a java application to implement currency converter (Dollar to BDT, EURO to BDT, Yen to BDT and vice versa), distance converter (meter to KM and vice versa), time converter (hours to minutes, seconds and vice versa) using package .

**SOLVE:**

import java.util.\*;

import java.text.\*;

import currencyconverter.\*;

import distanceconverter.\*;

import timeconverter.\*;

public class ConverterApp {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

CurrencyConverter cc = new CurrencyConverter();

DistanceConverter dc = new DistanceConverter();

TimeConverter tc = new TimeConverter();

System.out.println("Welcome to the Converter App");

System.out.println("Please choose one of the following options:");

System.out.println("1. Currency Converter");

System.out.println("2. Distance Converter");

System.out.println("3. Time Converter");

System.out.println("4. Exit");

int choice = input.nextInt();

switch (choice) {

case 1:

cc.currencyConverter();

break;

case 2:

dc.distanceConverter();

break;

case 3:

tc.timeConverter();

break;

case 4:

System.out.println("Thank you for using the Converter App");

System.exit(0);

break;

default:

System.out.println("Invalid option. Please try again.");

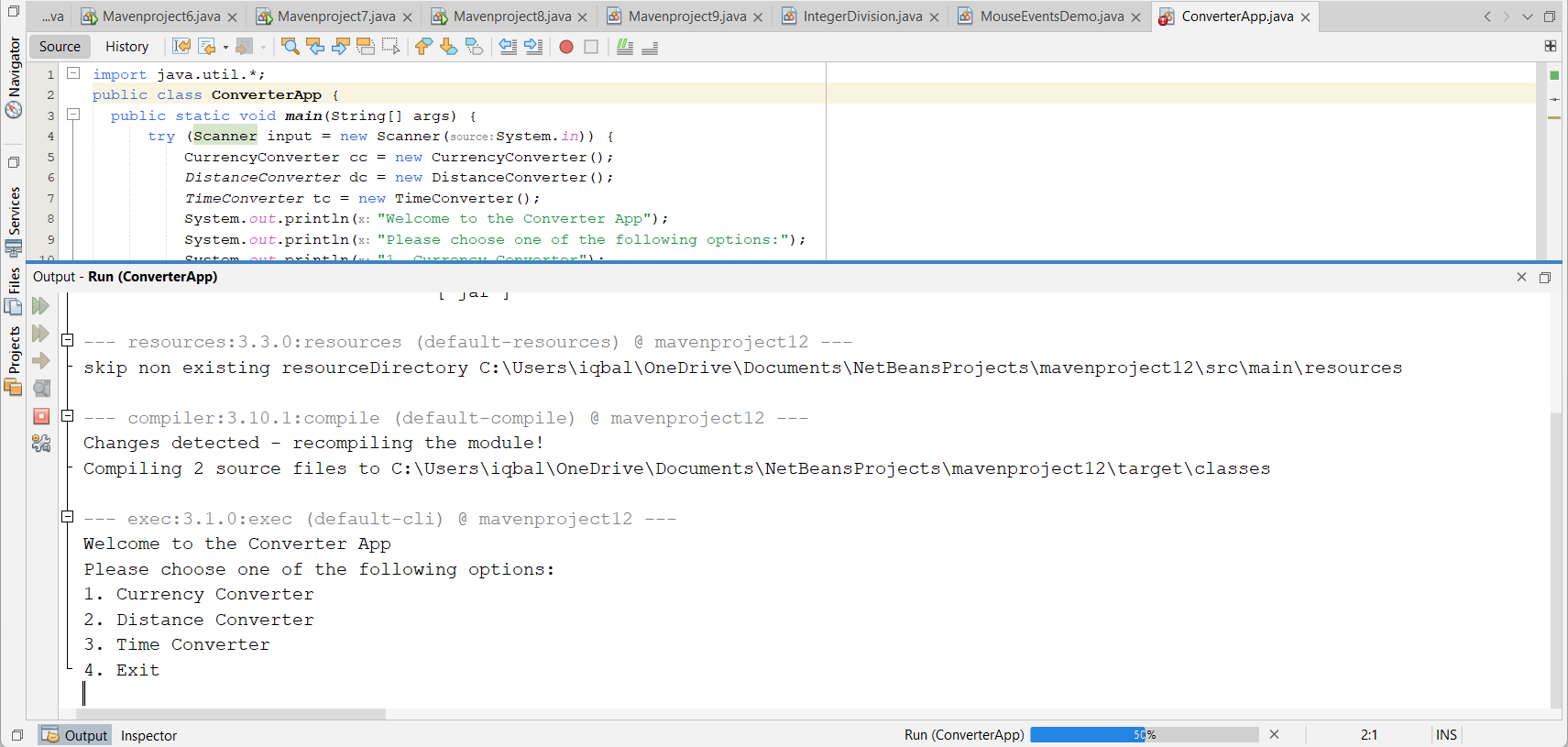
break;

}

input.close();

}

}



**QUESTION-11:** write a java program to implement Interface using extends keyword.

**SOLVE:**

interface Shape {

double calculateArea();

}

class Circle implements Shape {

private double radius;

public Circle(double radius) {

this.radius = radius;

}

public double calculateArea() {

return Math.PI \* radius \* radius;

}

}

class Rectangle implements Shape {

private double length;

private double width;

public Rectangle(double length, double width) {

this.length = length;

this.width = width;

}

public double calculateArea() {

return length \* width;

}

}

public class InterfaceExample {

public static void main(String[] args) {

Circle circle = new Circle(5);

Rectangle rectangle = new Rectangle(4, 6);

System.out.println("Area of Circle: " + circle.calculateArea());

System.out.println("Area of Rectangle: " + rectangle.calculateArea());

}

}

