**Q.1) Write a program to create two threads which shows messages Good morning and Good evening as per timing ?**

**Q.2) Write a program for example to try and catch block . In this check whether the given array size is negative or not .**

import java.util.Scanner;

public class ArraySizeChecker {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

try {

System.out.print("Enter the size of the array: ");

int size = scanner.nextInt();

if (size < 0) {

throw new NegativeArraySizeException("Array size cannot be negative.");

}

int[] array = new int[size];

System.out.println("Array of size " + size + " created successfully.");

} catch (NegativeArraySizeException e) {

System.out.println("Error: " + e.getMessage());

} catch (Exception e) {

System.out.println("Error: Invalid input.");

} finally {

scanner.close();

}

}

}

**Q.3) write a program for example of multiple catch statements occurring in a program** .

java.util.\*;

public class prac2 {

public static void main(String[] args) {

// TODO Auto-generated method stub

Scanner scn = new Scanner(System.in);

try

{

int n = Integer.parseInt(scn.nextLine());

if (99%n == 0)

System.out.println(n + " is a factor of 99");

}

catch (ArithmeticException ex)

{

System.out.println("Arithmetic " + ex);

}

catch (NumberFormatException ex)

{

System.out.println("Number Format Exception " + ex);

}

}

}

**Q.4) Write a program to illustrate sub class exception precedence over base class .**

class BaseException extends Exception {

public BaseException(String message) {

super(message);

}

}

class SubException extends BaseException {

public SubException(String message) {

super(message);

}

}

public class ExceptionPrecedence {

public static void main(String[] args) {

try {

throw new SubException("SubException is thrown");

} catch (SubException e) {

System.out.println("SubException caught: " + e.getMessage());

} catch (BaseException e) {

System.out.println("BaseException caught: " + e.getMessage());

}

}

}

**Q5) Write a program to illustrate usage of try/catch with the finally clause in java program.**

import java.io.\*;

public class TryCatchFinally {

public static void main(String[] args) {

BufferedReader reader = null;

try {

reader = new BufferedReader(new FileReader("input.txt"));

String line = null;

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

System.out.println(line);

}

} catch (FileNotFoundException e) {

System.err.println("File not found: " + e.getMessage());

} catch (IOException e) {

System.err.println("Error reading file: " + e.getMessage());

} finally {

try {

if (reader != null) {

reader.close();

}

} catch (IOException e) {

System.err.println("Error closing file: " + e.getMessage());

}

}

}

}

**Q.6) Write a program to describe usage of throws clause.**

import java.io.\*;

public class ThrowsExample {

public static void main(String[] args) throws IOException {

FileReader file = new FileReader("file.txt");

BufferedReader fileInput = new BufferedReader(file);

// Print first 3 lines of file

for (int i = 0; i < 3; i++) {

System.out.println(fileInput.readLine());

}

fileInput.close();

}

}

**Q.7) Write a program for creation of user defined exceptions.**

// Define a custom exception class

class MyException extends Exception {

public MyException(String errorMessage) {

super(errorMessage);

}

}

// Main class to demonstrate the custom exception

public class Main {

public static void main(String[] args) {

try {

// Throw the custom exception with a message

throw new MyException("This is a custom exception message");

} catch (MyException e) {

// Catch the custom exception and print the message

System.out.println("Caught MyException: " + e.getMessage());

}

}

}

**Q.8) Write a program to give the example for button control .**

import javax.swing.\*;

public class ButtonExample extends JFrame {

private JButton button;

public ButtonExample() {

super("Button Example");

// Create the button

button = new JButton("Click Me!");

// Add an action listener to handle button clicks

button.addActionListener(e -> {

JOptionPane.showMessageDialog(this, "You clicked the button!");

});

// Add the button to the frame

add(button);

// Configure the frame

setSize(300, 200);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLocationRelativeTo(null);

setVisible(true);

}

public static void main(String[] args) {

new ButtonExample();

}

}

**Q.9) Write a program to give the example for panel control**.

import javax.swing.\*;

import java.awt.\*;

public class PanelControlExample {

public static void main(String[] args) {

// Create a new JFrame and set its size

JFrame frame = new JFrame("Panel Control Example");

frame.setSize(300, 200);

// Create a new JPanel and set its background color

JPanel panel = new JPanel();

panel.setBackground(Color.WHITE);

// Create two new JLabels and add them to the panel

JLabel label1 = new JLabel("Label 1");

panel.add(label1);

JLabel label2 = new JLabel("Label 2");

panel.add(label2);

// Add the panel to the frame

frame.add(panel);

// Set the frame to be visible

frame.setVisible(true);

}

}

**Q.10) Write a program that will display check boxes and option buttons that are numbered from 1 to 5. Use a textbox to………………**

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class CheckboxRadioButtonExample extends JFrame implements ItemListener {

private JCheckBox[] checkBoxes;

private JRadioButton[] radioButtons;

private JTextField textField;

public CheckboxRadioButtonExample() {

// Set up the JFrame with a layout

super("Checkbox and Radio Button Example");

setLayout(new FlowLayout());

// Create and add checkboxes to the JFrame

checkBoxes = new JCheckBox[5];

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

checkBoxes[i] = new JCheckBox("Checkbox " + (i+1));

checkBoxes[i].addItemListener(this);

add(checkBoxes[i]);

}

// Create and add radio buttons to the JFrame

radioButtons = new JRadioButton[5];

ButtonGroup radioButtonGroup = new ButtonGroup();

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

radioButtons[i] = new JRadioButton("Radio Button " + (i+1));

radioButtons[i].addItemListener(this);

radioButtonGroup.add(radioButtons[i]);

add(radioButtons[i]);

}

// Create and add a text field to display the number of checked boxes or buttons

textField = new JTextField(10);

textField.setEditable(false);

add(textField);

// Set up the JFrame and make it visible

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setSize(300, 150);

setVisible(true);

}

// Handle checkbox and radio button events

public void itemStateChanged(ItemEvent e) {

int numChecked = 0;

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

if (checkBoxes[i].isSelected() || radioButtons[i].isSelected()) {

numChecked++;

}

}

textField.setText("Number checked: " + numChecked);

}

public static void main(String[] args) {

new CheckboxRadioButtonExample();

}

}

**Q.11) Write a program to create a simple calculator.**

import java.awt.\*;

import java.awt.event.\*;

class Calculator implements ActionListener {

// Declaring Objects

Frame f = new Frame();

Label l1 = new Label("First Number");

Label l2 = new Label("Second Number");

Label l3 = new Label("Result");

TextField t1 = new TextField();

TextField t2 = new TextField();

TextField t3 = new TextField();

Button b1 = new Button("Add");

Button b2 = new Button("Sub");

Button b3 = new Button("Mul");

Button b4 = new Button("Div");

Button b5 = new Button("Cancel");

Calculator() {

// Giving Coordinates

l1.setBounds(50, 100, 100, 20);

l2.setBounds(50, 140, 100, 20);

l3.setBounds(50, 180, 100, 20);

t1.setBounds(200, 100, 100, 20);

t2.setBounds(200, 140, 100, 20);

t3.setBounds(200, 180, 100, 20);

b1.setBounds(50, 250, 50, 20);

b2.setBounds(110, 250, 50, 20);

b3.setBounds(170, 250, 50, 20);

b4.setBounds(230, 250, 50, 20);

b5.setBounds(290, 250, 50, 20);

// Adding components to the frame

f.add(l1);

f.add(l2);

f.add(l3);

f.add(t1);

f.add(t2);

f.add(t3);

f.add(b1);

f.add(b2);

f.add(b3);

f.add(b4);

f.add(b5);

b1.addActionListener(this);

b2.addActionListener(this);

b3.addActionListener(this);

b4.addActionListener(this);

b5.addActionListener(this);

f.setLayout(null);

f.setVisible(true);

f.setSize(400, 350);

}

public void actionPerformed(ActionEvent e) {

int n1 = Integer.parseInt(t1.getText());

int n2 = Integer.parseInt(t2.getText());

if (e.getSource() == b1) {

t3.setText(String.valueOf(n1 + n2));

}

if (e.getSource() == b2) {

t3.setText(String.valueOf(n1 - n2));

}

if (e.getSource() == b3) {

t3.setText(String.valueOf(n1 \* n2));

}

if (e.getSource() == b4) {

t3.setText(String.valueOf(n1 / n2));

}

if (e.getSource() == b5) {

System.exit(0);

}

}

public static void main(String... s) {

new Calculator();

}

}

**Q.12) Write a program as above with a combo box and list boxes instead.**

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class Calculator implements ActionListener {

private JFrame frame;

private JTextField num1Field, num2Field, resultField;

private JComboBox<String> operatorComboBox;

private JList<String> historyList;

private DefaultListModel<String> historyListModel;

public Calculator() {

// Create components

JLabel num1Label = new JLabel("Number 1:");

num1Field = new JTextField(10);

JLabel num2Label = new JLabel("Number 2:");

num2Field = new JTextField(10);

JLabel operatorLabel = new JLabel("Operator:");

String[] operators = {"+", "-", "\*", "/"};

operatorComboBox = new JComboBox<>(operators);

JButton calculateButton = new JButton("Calculate");

calculateButton.addActionListener(this);

JLabel resultLabel = new JLabel("Result:");

resultField = new JTextField(10);

resultField.setEditable(false);

JLabel historyLabel = new JLabel("History:");

historyListModel = new DefaultListModel<>();

historyList = new JList<>(historyListModel);

JScrollPane historyScrollPane = new JScrollPane(historyList);

historyScrollPane.setPreferredSize(new Dimension(200, 100));

// Create layout

JPanel inputPanel = new JPanel(new GridLayout(3, 2, 10, 10));

inputPanel.add(num1Label);

inputPanel.add(num1Field);

inputPanel.add(num2Label);

inputPanel.add(num2Field);

inputPanel.add(operatorLabel);

inputPanel.add(operatorComboBox);

JPanel buttonPanel = new JPanel();

buttonPanel.add(calculateButton);

JPanel resultPanel = new JPanel(new GridLayout(2, 2, 10, 10));

resultPanel.add(resultLabel);

resultPanel.add(resultField);

resultPanel.add(historyLabel);

resultPanel.add(historyScrollPane);

JPanel mainPanel = new JPanel(new BorderLayout());

mainPanel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));

mainPanel.add(inputPanel, BorderLayout.CENTER);

mainPanel.add(buttonPanel, BorderLayout.SOUTH);

mainPanel.add(resultPanel, BorderLayout.EAST);

// Create frame

frame = new JFrame("Calculator");

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setContentPane(mainPanel);

frame.pack();

frame.setVisible(true);

}

public void actionPerformed(ActionEvent e) {

// Get input values

double num1 = Double.parseDouble(num1Field.getText());

double num2 = Double.parseDouble(num2Field.getText());

String operator = (String) operatorComboBox.getSelectedItem();

// Perform calculation and update result

double result;

switch (operator) {

case "+":

result = num1 + num2;

break;

case "-":

result = num1 - num2;

break;

case "\*":

result = num1 \* num2;

break;

case "/":

result = num1 / num2;

break;

default:

result = 0;

break;

}

resultField.setText(Double.toString(result));

// Add calculation to history

String historyItem = num1 + " " + operator + " " + num2 + " = " + result;

historyListModel.addElement(historyItem);

}

public static void main(String[] args) {

new Calculator();

}

}

**Q.13) Write a program that displays the x and y position of the cursour movement using mouse** .

import java.applet.\*;

import java.awt.\*;

import java.awt.event.\*;

public class mouseevent2 extends Frame

{

int x=0;

int y=0;

public void init()

{

addMouseListener(new mymouselistener());

addWindowListener(new MyWindowAdapter());

}

public void paint(Graphics g)

{

g.drawLine(x,y,x,y);

g.drawString(x + ","+ y, x,y);

}

public class mymouselistener extends MouseAdapter

{

public void mouseClicked(MouseEvent e)

{

x = e.getX();

y = e.getY();

repaint();

}

}

class MyWindowAdapter extends WindowAdapter

{

public void windowClosing(WindowEvent we)

{

System.exit(0);

}

}

public static void main(String p[])

{

mouseevent2 me2 = new mouseevent2();

me2.setSize(250, 250);

me2.setVisible(true);

me2.init();

}

}

**Q.14) Write a program to create a canvas .**

import java.awt.\*;

// class to construct a frame and containing main method

public class CanvasExample

{

// class constructor

public CanvasExample()

{

// creating a frame

Frame f = new Frame("Canvas Example");

// adding canvas to frame

f.add(new MyCanvas());

// setting layout, size and visibility of frame

f.setLayout(null);

f.setSize(400, 400);

f.setVisible(true);

}

// main method

public static void main(String args[])

{

new CanvasExample();

}

}

// class which inherits the Canvas class

// to create Canvas

class MyCanvas extends Canvas

{

// class constructor

public MyCanvas() {

setBackground (Color.GRAY);

setSize(300, 200);

}

// paint() method to draw inside the canvas

public void paint(Graphics g)

{

// adding specifications

g.setColor(Color.red);

g.fillOval(75, 75, 150, 75);

}

}

**Q.15) Write a program that displays the x and y position of the cursour movement using keyboard.**

import java.awt.\*;

import java.awt.event.\*;

public class CursorPosition implements MouseMotionListener {

public static void main(String[] args) {

Frame frame = new Frame("Cursor Position");

frame.addMouseMotionListener(new CursorPosition());

frame.setSize(300, 200);

frame.setVisible(true);

}

public void mouseMoved(MouseEvent e) {

int x = e.getX();

int y = e.getY();

System.out.println("Cursor position: (" + x + ", " + y + ")");

}

public void mouseDragged(MouseEvent e) {}

}

**Q.16) Write a program to create a text box control .**

import javax.swing.\*;

public class TextBoxControlExample {

public static void main(String[] args) {

JFrame frame = new JFrame("Text Box Control Example");

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

// create a text box control

JTextField textField = new JTextField();

textField.setBounds(50, 50, 200, 30);

// add the text box control to the frame

frame.add(textField);

// set the frame size and make it visible

frame.setSize(300, 150);

frame.setLayout(null);

frame.setVisible(true);

}

}

**Q.17) Write a program to create an analog clock .**

**Q.18) Write a program for temperature converter.**

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class TemperatureConverter extends JFrame implements ActionListener {

private JLabel celsiusLabel, fahrenheitLabel, resultLabel;

private JTextField celsiusField, fahrenheitField, resultField;

private JButton convertButton;

public TemperatureConverter() {

setTitle("Temperature Converter");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setSize(400, 200);

celsiusLabel = new JLabel("Celsius:");

fahrenheitLabel = new JLabel("Fahrenheit:");

resultLabel = new JLabel("Result:");

celsiusField = new JTextField(10);

fahrenheitField = new JTextField(10);

resultField = new JTextField(10);

resultField.setEditable(false);

convertButton = new JButton("Convert");

convertButton.addActionListener(this);

setLayout(new GridLayout(4, 2));

add(celsiusLabel);

add(celsiusField);

add(fahrenheitLabel);

add(fahrenheitField);

add(resultLabel);

add(resultField);

add(convertButton);

setVisible(true);

}

public void actionPerformed(ActionEvent e) {

if (e.getSource() == convertButton) {

if (!celsiusField.getText().isEmpty()) {

double celsius = Double.parseDouble(celsiusField.getText());

double fahrenheit = (celsius \* 9 / 5) + 32;

fahrenheitField.setText(String.format("%.2f", fahrenheit));

} else if (!fahrenheitField.getText().isEmpty()) {

double fahrenheit = Double.parseDouble(fahrenheitField.getText());

double celsius = (fahrenheit - 32) \* 5 / 9;

celsiusField.setText(String.format("%.2f", celsius));

}

}

}

public static void main(String[] args) {

new TemperatureConverter();

}

}

**Q.19) Write a program to calculate the area of circle and rectangle using AWT and GUI .**

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class AreaCalculator extends JFrame implements ActionListener {

private JLabel radiusLabel, lengthLabel, widthLabel, resultLabel;

private JTextField radiusField, lengthField, widthField, resultField;

private JButton circleButton, rectangleButton;

public AreaCalculator() {

setTitle("Area Calculator");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setSize(400, 200);

radiusLabel = new JLabel("Radius:");

lengthLabel = new JLabel("Length:");

widthLabel = new JLabel("Width:");

resultLabel = new JLabel("Result:");

radiusField = new JTextField(10);

lengthField = new JTextField(10);

widthField = new JTextField(10);

resultField = new JTextField(10);

circleButton = new JButton("Calculate Circle Area");

rectangleButton = new JButton("Calculate Rectangle Area");

circleButton.addActionListener(this);

rectangleButton.addActionListener(this);

setLayout(new GridLayout(4, 2));

add(radiusLabel);

add(radiusField);

add(lengthLabel);

add(lengthField);

add(widthLabel);

add(widthField);

add(resultLabel);

add(resultField);

add(circleButton);

add(rectangleButton);

setVisible(true);

}

public void actionPerformed(ActionEvent e) {

if (e.getSource() == circleButton) {

double radius = Double.parseDouble(radiusField.getText());

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

resultField.setText(String.format("%.2f", area));

} else if (e.getSource() == rectangleButton) {

double length = Double.parseDouble(lengthField.getText());

double width = Double.parseDouble(widthField.getText());

double area = length \* width;

resultField.setText(String.format("%.2f", area));

}

}

public static void main(String[] args) {

new AreaCalculator();

}

}

**Q.20) Write a program to create a simple “Registration form “ of a student .**

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class StudentRegistrationForm extends JFrame implements ActionListener {

private JLabel nameLabel, ageLabel, genderLabel, courseLabel, resultLabel;

private JTextField nameField, ageField;

private JComboBox<String> genderBox, courseBox;

private JButton submitButton;

public StudentRegistrationForm() {

setTitle("Student Registration Form");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setSize(400, 200);

nameLabel = new JLabel("Name:");

ageLabel = new JLabel("Age:");

genderLabel = new JLabel("Gender:");

courseLabel = new JLabel("Course:");

resultLabel = new JLabel();

nameField = new JTextField(10);

ageField = new JTextField(3);

String[] genders = {"Male", "Female", "Other"};

genderBox = new JComboBox<>(genders);

String[] courses = {"Computer Science", "Engineering", "Business", "Arts"};

courseBox = new JComboBox<>(courses);

submitButton = new JButton("Submit");

submitButton.addActionListener(this);

setLayout(new GridLayout(5, 2));

add(nameLabel);

add(nameField);

add(ageLabel);

add(ageField);

add(genderLabel);

add(genderBox);

add(courseLabel);

add(courseBox);

add(resultLabel);

add(submitButton);

setVisible(true);

}

public void actionPerformed(ActionEvent e) {

String name = nameField.getText();

int age = Integer.parseInt(ageField.getText());

String gender = (String) genderBox.getSelectedItem();

String course = (String) courseBox.getSelectedItem();

resultLabel.setText("Registration successful:\nName: " + name +

"\nAge: " + age +

"\nGender: " + gender +

"\nCourse: " + course);

}

public static void main(String[] args) {

new StudentRegistrationForm();

}

}

***THE END .***