Rayat Shikshan Sanstha's

KARMAVEER BHAURAO PATIL COLLEGE, VASHI

[Autonomous College]

Reaccredited by NAAC with Grade 'A+' (CGPA 3.53) | ISO 9001: 2008 Certified Institute

'Best College' Award by University of Mumbai

Name - Arin Nitin Mandre

Roll No - 243319

Subject - Core Java

INDEX

SR.NO	TITLE	DATE	SIGN
1	Write a program to find the average and sum ofthe Numbers Using Command line argument.		
2	Write a program to demonstrate Type Casting.		
3	Write a program to design a class account using the inheritance and static that show all functions ofthe bank (withdrawal, deposit).		
4	Write a program to design a class using abstractMethods and Classes.		
5	Write a program to handle the Exception using tryand multiple catch blocks.		
6	Write a program to Create a package that accessesthe member of the external class as well as the same package.		
7	Write a program to create a thread thatimplements the Runnable interface.		
8	AWT and SWING : (i) Write a program to create a form of ATM byusing AWT. (ii) Write a program to create a form by usingswing.		

PRACTICAL NO - 1

- **Write a program to find the average and sum ofthe Numbers Using Command line argument.**

CODE:

```
public class CalculateSumAndAverage {
    public static void main(String[] args) {

        if (args.length == 0) {
            System.out.println("Please provide numbers as command-line arguments.");
            return;
        }

        double sum = 0.0;

        try {

            for (String arg : args) {
                sum += Double.parseDouble(arg);
            }
        } catch (NumberFormatException e) {
            System.out.println("Error: Please ensure all command-line arguments are valid numbers.");
            return;
        }


        double average = sum / args.length;


        System.out.println("Sum of numbers: " + sum);
        System.out.println("Average of numbers: " + average);
    }
}
```

OUTPUT:

```
Sum of numbers: 60.0
Average of numbers: 20.0
```

PRACTICAL NO - 2

- **Write a program to demonstrate Type Casting.**

  1. Narrow type casting.

CODE:

```
public class NarrowingTypeCasting {
    public static void main(String[] args) {
        double d = 12.44;
        long l = (long) d;
        int i = (int) l;

        System.out.println("Before conversion, the double value: " + d);
        System.out.println("After conversion to long value: " + l);
        System.out.println("After conversion to int value: " + i);
    }
}
```

OUTPUT:

```
Before conversion, the double value: 12.44
After conversion to long value: 12
After conversion to int value: 12
```

2. Widening type casting

CODE:

```
public class WideningTypeCasting {
    public static void main(String[] args) {
        int a = 5;
        long b = a;
        double d = b;

        System.out.println("Before conversion, the int value: " + a);
        System.out.println("After conversion to long value: " + b);
        System.out.println("After conversion to double value: " + d);
    }
}
```

OUTPUT:

```
Before conversion, the int value: 5
After conversion to long value: 5
After conversion to double value: 5.0
```

PRACTICAL NO - 3

- **Write a program to design a class account using the inheritance and static that show all functions ofthe bank (withdrawal, deposit).**

CODE:

```
public class Account {
    private String accountHolderName;
    private double balance;

    public Account(String name, double initialBalance) {
        accountHolderName = name;
        balance = initialBalance;
    }

    public void deposit(double amount) {
        balance += amount;
    }

    public void withdraw(double amount) {
        balance -= amount;
    }

    public void displayAccountInfo() {
        System.out.println("Account Holder Name: " + accountHolderName);
        System.out.println("Balance: " + balance);
    }

    public static void main(String[] args) {
        Account account = new Account("Arin", 1000.0);
        account.displayAccountInfo();
        account.deposit(500.0);
        account.displayAccountInfo();
        account.withdraw(200.0);
        account.displayAccountInfo();
    }
}
```

OUTPUT:

```
Account Holder Name: Arin
Balance: 1000.0
Account Holder Name: Arin
Balance: 1500.0
Account Holder Name: Arin
Balance: 1300.0
```

PRACTICAL NO - 4

- **Write a program to design a class using abstractMethods and Classes.**

CODE:

```
class StudentName {
    String name;

    StudentName(String name) {
        this.name = name;
    }

    void displayInfo() {
        System.out.println("Student name: " + name);
    }
}

class Student extends StudentName {
    int rollNumber;

    Student(String name, int rollNumber) {
        super(name);
        this.rollNumber = rollNumber;
    }

    void displayInfo() {
        super.displayInfo();
        System.out.println("Roll Number: " + rollNumber);
    }
}

public class Main {
    public static void main(String[] args) {
        Student student = new Student("Arin", 1234);
        student.displayInfo();
    }
}
```

OUTPUT:

```
Student name: Arin
Roll Number: 1234
```

PRACTICAL NO - 5

- **Write a program to handle the Exception using tryand multiple catch blocks.**

CODE:

```
public class ErrorHandling {
    public static void main(String[] args) {
        try {
            int num1 = 10;
            int num2 = 0;

            int result = num1 / num2;
            System.out.println("Result: " + result);

            String str = null;

            System.out.println(str.length());

            int[] arr = new int[5];

            System.out.println(arr[10]);
        } catch (ArithmeticException e) {
            System.out.println("Arithmetic Error: " + e.getMessage());
        } catch (NullPointerException e) {
            System.out.println("Null Pointer Error: " + e.getMessage());
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Array Index Error: " + e.getMessage());
        } catch (Exception e) {
            System.out.println("Unknown Error: " + e.getMessage());
        }
    }
}
```

OUTPUT:

```
Arithmetic Error: / by zero
```

PRACTICAL NO - 6

- **Write a program to Create a package that accessesthe member of the external class as well as the same package.**

CODE:

```
package mypackage;

class ExternalClass {
    public static void externalMethod() {
        System.out.println("This is an external method");
    }
}

class MyOtherClass {
    public static void myMethod() {
        System.out.println("This is a method in the same package");
    }
}

public class MyClass {
    public static void main(String[] args) {
        ExternalClass.externalMethod();
        MyOtherClass.myMethod();
    }
}
```

OUTPUT:

```
This is an external method
This is a method in the same package
```

PRACTICAL NO - 7

- **Write a program to create a thread thatimplements the Runnable interface.**

CODE:

```
public class CustomRunnableThread implements Runnable {
    @Override
    public void run() {
        for (int i = 0; i < 5; i++) {
            System.out.println("Thread is running... " + i);
        }
    }

    public static void main(String[] args) {
        CustomRunnableThread customThread = new CustomRunnableThread();
        Thread thread = new Thread(customThread);
        thread.start();
    }
}
```

OUTPUT:

```
Thread is running... 0
Thread is running... 1
Thread is running... 2
Thread is running... 3
Thread is running... 4
```

PRACTICAL NO - 8

AWT and SWING :

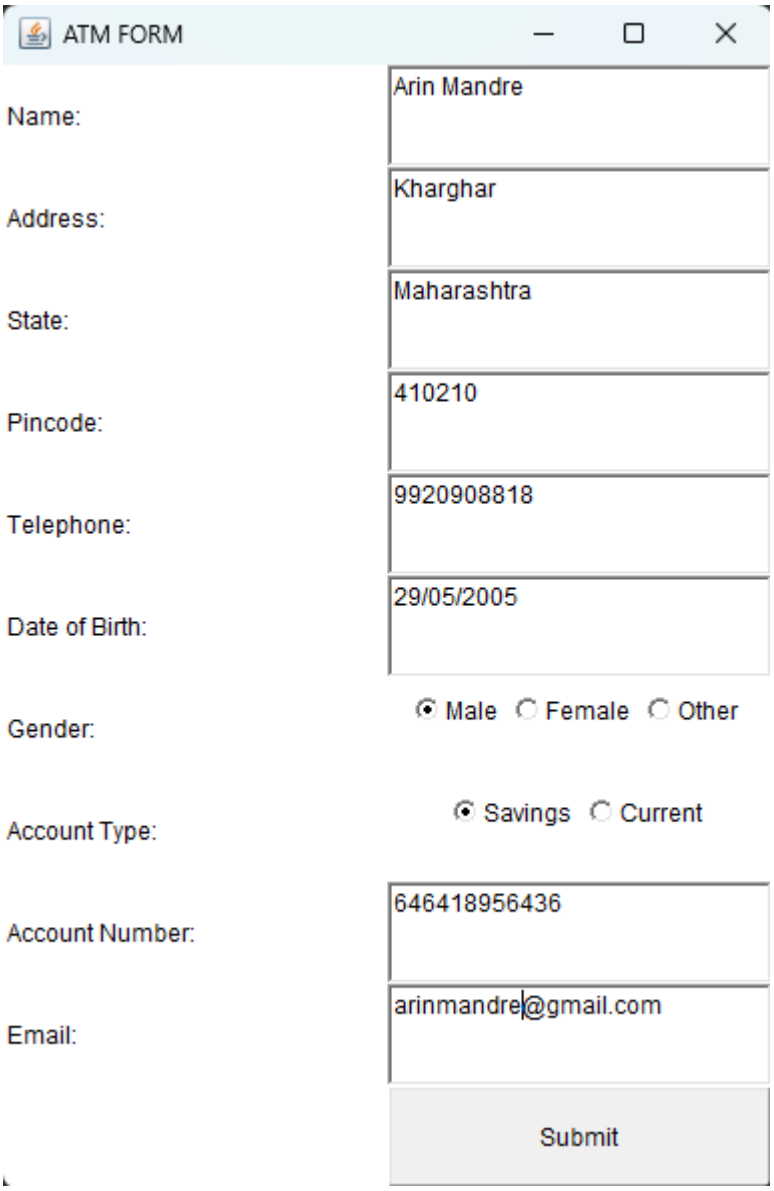
1. Write a program to create a form of ATM byusing AWT.

```
import java.awt.*;
import java.awt.event.*;

public class ATMForm {
    public static void main(String[] args) {

        Frame frame = new Frame("ATM FORM");
        frame.setSize(400, 600);
        frame.setLayout(new GridLayout(0, 2));


        Label nameLabel = new Label("Name:");
        TextField nameField = new TextField(20);
        frame.add(nameLabel);
        frame.add(nameField);


        Label addressLabel = new Label("Address:");
        TextField addressField = new TextField(20);
        frame.add(addressLabel);
        frame.add(addressField);


        Label stateLabel = new Label("State:");
        TextField stateField = new TextField(20);
        frame.add(stateLabel);
        frame.add(stateField);


        Label pincodeLabel = new Label("Pincode:");
        TextField pincodeField = new TextField(20);
        frame.add(pincodeLabel);
        frame.add(pincodeField);


        Label telephoneLabel = new Label("Telephone:");
        TextField telephoneField = new TextField(20);
        frame.add(telephoneLabel);
        frame.add(telephoneField);


        Label dobLabel = new Label("Date of Birth:");
        TextField dobField = new TextField(20);
        frame.add(dobLabel);
        frame.add(dobField);
```

```
    Label genderLabel = new Label("Gender:");
    CheckboxGroup genderGroup = new CheckboxGroup();
    Checkbox maleCheckbox = new Checkbox("Male", genderGroup, false);
    Checkbox femaleCheckbox = new Checkbox("Female", genderGroup, false);
    Checkbox otherCheckbox = new Checkbox("Other", genderGroup, false);

    Panel genderPanel = new Panel();
    genderPanel.add(maleCheckbox);
    genderPanel.add(femaleCheckbox);
    genderPanel.add(otherCheckbox);

    frame.add(genderLabel);
    frame.add(genderPanel);


    Label accountTypeLabel = new Label("Account Type:");
    CheckboxGroup accountTypeGroup = new CheckboxGroup();
    Checkbox savingsCheckbox = new Checkbox("Savings", accountTypeGroup, false);
    Checkbox currentCheckbox = new Checkbox("Current", accountTypeGroup, false);

    Panel accountTypePanel = new Panel();
    accountTypePanel.add(savingsCheckbox);
    accountTypePanel.add(currentCheckbox);

    frame.add(accountTypeLabel);
    frame.add(accountTypePanel);


    Label accountNumberLabel = new Label("Account Number:");
    TextField accountNumberField = new TextField(20);
    frame.add(accountNumberLabel);
    frame.add(accountNumberField);


    Label emailLabel = new Label("Email:");
    TextField emailField = new TextField(20);
    frame.add(emailLabel);
    frame.add(emailField);


    Button submitButton = new Button("Submit");
    frame.add(new Label());
    frame.add(submitButton);
```

```
        submitButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                System.out.println("Form Submitted");
                System.out.println("Name: " + nameField.getText());
                System.out.println("Address: " + addressField.getText());
                System.out.println("State: " + stateField.getText());
                System.out.println("Pincode: " + pincodeField.getText());
                System.out.println("Telephone: " + telephoneField.getText());
                System.out.println("DOB: " + dobField.getText());
                System.out.println("Gender: " + genderGroup.getSelectedCheckbox().getLabel());
                System.out.println("Account Type: " + accountTypeGroup.getSelectedCheckbox().getLabel());
                System.out.println("Account Number: " + accountNumberField.getText());
                System.out.println("Email: " + emailField.getText());
            }
        });


        frame.setVisible(true);
    }
}
```

OUTPUT:


ATM FORM
Name:
Arin Mandre
Address:
Kharghar
State:
Maharashtra
Pincode:
410210
Telephone:
9920908818
Date of Birth:
29/05/2005
Gender:
Male
Female
Other
Account Type:
Savings
Current
Account Number:
646418956436
Email:
arinmandre@gmail.com
Submit

2. **Write a program to create a form by usingswing.**

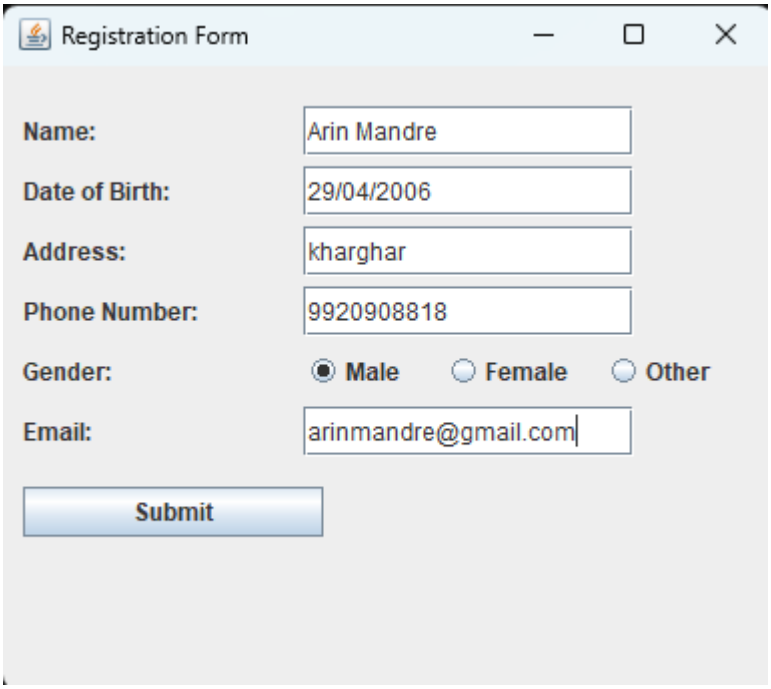
CODE:

```
package pom.mycompany.registerform;

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class RegisterForm {
    public static void main(String[] args) {

        JFrame f = new JFrame("Registration Form");
        f.setSize(400, 350);
        f.setLayout(null);


        JLabel nameLabel = new JLabel("Name:");
        nameLabel.setBounds(10, 20, 50, 25);
        f.add(nameLabel);
        JTextField nameText = new JTextField(20);
        nameText.setBounds(150, 20, 165, 25);
        f.add(nameText);


        JLabel dobLabel = new JLabel("Date of Birth:");
        dobLabel.setBounds(10, 50, 100, 25);
        f.add(dobLabel);
        JTextField dobText = new JTextField(20);
        dobText.setBounds(150, 50, 165, 25);
        f.add(dobText);


        JLabel addressLabel = new JLabel("Address:");
        addressLabel.setBounds(10, 80, 80, 25);
        f.add(addressLabel);
        JTextField addressText = new JTextField(20);
        addressText.setBounds(150, 80, 165, 25);
        f.add(addressText);


        JLabel phoneLabel = new JLabel("Phone Number:");
        phoneLabel.setBounds(10, 110, 100, 25);
        f.add(phoneLabel);
        JTextField phoneText = new JTextField(20);
        phoneText.setBounds(150, 110, 165, 25);
        f.add(phoneText);
```

```
    JLabel genderLabel = new JLabel("Gender:");
    genderLabel.setBounds(10, 140, 80, 25);
    f.add(genderLabel);
    JRadioButton maleRadio = new JRadioButton("Male");
    maleRadio.setBounds(150, 140, 70, 25);
    JRadioButton femaleRadio = new JRadioButton("Female");
    femaleRadio.setBounds(220, 140, 80, 25);
    JRadioButton otherRadio = new JRadioButton("Other");
    otherRadio.setBounds(300, 140, 70, 25);


    ButtonGroup genderGroup = new ButtonGroup();
    genderGroup.add(maleRadio);
    genderGroup.add(femaleRadio);
    genderGroup.add(otherRadio);

    f.add(maleRadio);
    f.add(femaleRadio);
    f.add(otherRadio);


    JLabel emailLabel = new JLabel("Email:");
    emailLabel.setBounds(10, 170, 80, 25);
    f.add(emailLabel);
    JTextField emailText = new JTextField(20);
    emailText.setBounds(150, 170, 165, 25);
    f.add(emailText);


    JButton submitButton = new JButton("Submit");
    submitButton.setBounds(10, 210, 150, 25);
    f.add(submitButton);
```

```
        submitButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {

                System.out.println("Form Submitted");
                System.out.println("Name: " + nameText.getText());
                System.out.println("Date of Birth: " + dobText.getText());
                System.out.println("Address: " + addressText.getText());
                System.out.println("Phone Number: " + phoneText.getText());
                System.out.println("Gender: " + (maleRadio.isSelected() ? "Male" :
                                                femaleRadio.isSelected() ? "Female" : "Other"));
                System.out.println("Email: " + emailText.getText());


                nameText.setText("");
                dobText.setText("");
                addressText.setText("");
                phoneText.setText("");
                genderGroup.clearSelection();
                emailText.setText("");
            }
        });


        f.setVisible(true);
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```

CODE:


Registration Form
Name:
Arin Mandre
Date of Birth:
29/04/2006
Address:
kharghar
Phone Number:
9920908818
Gender:
Male
Female
Other
Email:
arinmandre@gmail.com
Submit

* * * * *