**JAVA FINAL**

1. **Create a basic text editor where the user can where the user can append and delete character from a string. Use stringbuffer class for this operation.**

**Ans:-**

import java.util.\*;

class TextEditor {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter the first string: ");

StringBuffer sb1 = new StringBuffer(sc.nextLine());

System.out.println("First StringBuffer: " + sb1);

System.out.print("Enter the second string: ");

StringBuffer sb2 = new StringBuffer(sc.nextLine());

sb1.append(sb2);

System.out.println("After appending: " + sb1);

sb1.delete(2, 4);

System.out.println("After deletion (index 2 to 4): " + sb1);

sc.close();

}

}

1. **Write a program that accepts two numbers as command-line arguments, calculates their sum.**

**Ans:-**

public class CommandLine {

public static void main(String[] args) {

if (args.length < 2) {

System.out.println("Provide two numbers as arguments.");

return;

}

try {

System.out.println("The sum is: " + (Integer.parseInt(args[0]) + Integer.parseInt(args[1])));

} catch (NumberFormatException e) {

System.out.println("Arguments must be integers.");

}

}

}

1. **A voter registration system checks if a person is 18 or older. If yes, they are eligible to vote; otherwise, they are not. Write a program using if-else.**

**Ans:-**

import java.util.Scanner;

public class VoterEligibility {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter your age: ");

int age = scanner.nextInt();

if (age >= 18) {

System.out.println("You are eligible to vote.");

} else {

System.out.println("You are not eligible to vote.");

}

scanner.close();

}

}

1. **A bank system needs to reverse a transaction ID for secure processing. Write a program to reverse the digits of a number (e.g., 1234 → 4321) using a do-while loop.**

**Ans:-**

import java.util.Scanner;

public class ReverseNO {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int number = sc.nextInt();

int reverseno = 0;

do {

reverseno = reverseno \* 10 + number % 10;

number /= 10;

} while (number != 0);

System.out.println("Reversed number is: " + reverseno);

sc.close();

}

}

1. **An event manager wants to display all even-numbered seats (1–20) for a special seating arrangement, skipping odd-numbered seats. Write a program to achieve this using a continue statement.**

**Ans:-**

public class Continue

{

public static void main(String [] arg)

{

for (int i=1;i<=20;i++)

{

if(i%2!=0)

{

continue;

}

System.out.println("Available Seats is "+ i );

}

}

}

\*JAVA CODE:=

1. Write a program that accepts two numbers as command-line arguments, calculates their sum:-

=>

import java.util.Scanner; class Sumofcal {

public static void main(String[] args) { if (args.length != 2) {

System.*out*.println("Enter Two Numbers"); return;

}

try {

int num1 = Integer.*parseInt*(args[0]); int num2 = Integer.*parseInt*(args[1]); int sum = num1 + num2;

System.*out*.println("The sum is: " + sum);

} catch (NumberFormatException e) { System.*out*.println("Invalid input. Please enter valid integers.");

}

}

}

Steps to run:-

* 1. **Save the Program**

Save the file as Sumofcal.java.

* 1. **Compile the Program**

javac Sumofcal.java

* 1. **Run the Program with Arguments**

Use the following syntax: java SumCalculator 5 10

1. A voter registration system checks if a person is 18 or older. If yes, they are eligible to vote; otherwise, they are not. Write a program using if-else:-

=>

import java.util.Scanner; public class Voter {

public static void main(String[] args) { Scanner sc = new Scanner(System.in);

// Ask for the user's age System.out.print("Enter your age: "); int age = sc.nextInt();

// Check if the user is eligible to vote if (age >= 18) {

System.out.println("You are eligible to vote.");

} else {

System.out.println("You are not eligible to vote.");

}

}

}

1. A restaurant has a menu where each item has a unique code (e.g., 1 for Pizza, 2 for Burger, 3 for Pasta). Write a program that takes the code as input and displays the corresponding item name using a switch statement:

=>

import java.util.Scanner; public class RestaurantMenu {

public static void main(String[] args) { Scanner sc = new Scanner(System.in);

// Display Menu System.out.println("Restaurant Menu:"); System.out.println("1. Pizza"); System.out.println("2. Burger"); System.out.println("3. Pasta");

System.out.println("Enter the item code (1-3): ");

// Take user input for the item code int code = sc.nextInt();

// Display item based on the code using switch switch (code) {

case 1:

System.out.println("You selected: Pizza"); break;

case 2:

System.out.println("You selected: Burger"); break;

case 3:

System.out.println("You selected: Pasta"); break;

default:

System.out.println("Invalid code! Please select a valid menu item (1-3).");

}

}

}

1. bank wants to check if a customers balance is below 500, and if so, apply a penalty fee of 10 otherwise, no fee. Write a program using the ternary operator:

=>

import java.util.Scanner;

public class BankPenalty {

public static void main(String[] args) { Scanner sc = new Scanner(System.in);

// Ask for the customer's balance System.out.print("Enter your account balance: "); double balance = sc.nextDouble();

// Apply penalty using ternary operator

balance = (balance < 500) ? (balance - 10) : balance;

// Display the result if (balance < 500) {

System.out.println("A penalty of $10 has been applied.");

} else {

System.out.println("No penalty applied.");

}

System.out.println("Your final balance is: $" + balance);

}

}

1. A student scored 45, 50, 24, 28, and 55 in 5 subjects. Out of 60 Write a program to calculate the average marks using arithmetic operators:

=>

public class AverageMarks {

public static void main(String[] args) {

// Declare and initialize marks for 5 subjects int subject1 = 45;

int subject2 = 50; int subject3 = 24; int subject4 = 28; int subject5 = 55;

// Calculate the total marks

int totalMarks = subject1 + subject2 + subject3 + subject4 + subject5;

// Calculate the average marks

double averageMarks = totalMarks / 5.0; // Use 5.0 to ensure floating-point division

// Display the results

System.out.println("Marks in 5 subjects: 45, 50, 24, 28, 55"); System.out.println("Total Marks: " + totalMarks); System.out.println("Average Marks: " + averageMarks);

}

}

1. Write a program for command line age difference calculator. Difference between age of person A and person B:

=>

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

if (args.length != 2) {

System.out.println("Please provide the ages of two people as command-line arguments."); return;

}

try {

int ageA = Integer.parseInt(args[0]); int ageB = Integer.parseInt(args[1]);

// Calculate the age difference

int ageDifference = Math.abs(ageA - ageB); // Use abs to ensure positive difference

// Display the result

System.out.println("Age of Person A: " + ageA); System.out.println("Age of Person B: " + ageB); System.out.println("The age difference is: " + ageDifference + " years");

} catch (NumberFormatException e) {

System.out.println("Invalid input. Please enter valid integer ages.");

}

}

}

\*Unit-2:

1. Write a program to print the first 10 natural numbers using a for loop:

=>

public class NaturalNumbers {

public static void main(String[] args) { System.out.println("The first 10 natural numbers are:");

// Use a for loop to print numbers from 1 to 10 for (int i = 1; i <= 10; i++) {

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

}

}

}

1. Write a program to calculate the factorial of a number using a while loop:

=>

import java.util.Scanner;

public class FactorialCalculator {

public static void main(String[] args) { Scanner sc = new Scanner(System.in);

// Ask user for a number System.out.print("Enter a number: "); int number = sc.nextInt();

if (number < 0) {

System.out.println("Factorial is not defined for negative numbers.");

} else {

int factorial = 1; // Initialize factorial

int i = number; // Start from the given number

// Calculate factorial using while loop while (i > 0) {

factorial \*= i; // Multiply current value of i with factorial i--; // Decrement i

}

// Display the result

System.out.println("The factorial of " + number + " is: " + factorial);

}

}

}

1. A bank system needs to reverse a transaction ID for secure processing. Write a program to reverse the digits of a number (e.g., 1234 → 4321) using a do-while loop:

=>

import java.util.Scanner;

public class ReverseNumber {

public static void main(String[] args) { Scanner sc = new Scanner(System.in);

// Ask for the transaction ID System.out.print("Enter the transaction ID: "); int number = sc.nextInt();

// Validate input if (number < 0) {

System.out.println("Please enter a positive number.");

} else {

int reversed = 0;

// Reverse the number using do-while loop do {

int digit = number % 10; reversed = reversed \* 10 + digit; number /= 10;

} while (number > 0);

// Display the result

System.out.println("Reversed Transaction ID: " + reversed);

}

}

}

1. An online shopping platform wants to find the first item in a search result matching the user's query and stop further processing. Write a program to find the first occurrence of a number in an array and terminate the loop using a break statement:

=>

import java.util.Scanner;

public class FirstOccurrence {

public static void main(String[] args) { Scanner sc = new Scanner(System.in);

// Sample array representing search results (item IDs) int[] items = {101, 205, 307, 410, 512, 205, 678};

// Ask the user for the search query (number to find) System.out.print("Enter the item ID to search: "); int query = sc.nextInt();

// Find the first occurrence of the query in the array boolean found = false;

for (int i = 0; i < items.length; i++) { if (items[i] == query) {

System.out.println("Item found at index: " + i); found = true;

break; // Stop further processing after the first match

}

}

if (!found) {

System.out.println("Item not found in the search results.");

}

}

}

1. Scenario-Based Question:

An event manager wants to display all even-numbered seats (1–20) for a special seating arrangement, skipping odd-numbered seats. Write a program to achieve this using a continue statement:

=>

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

System.out.println("Even-numbered seats (1–20) for the special seating :");

// Loop through seat numbers from 1 to 20 for (int seat = 1; seat <= 20; seat++) {

// Skip odd-numbered seats using continue if (seat % 2 != 0) {

continue; // Skip to the next iteration if the seat is odd

}

// Display even-numbered seat System.out.print(seat + " ");

}

}

}

1. Create a simple ATM-like system. Assume the user has an initial balance of $500. The program should offer the user the option to withdraw, deposit, or check the balance. After each transaction, print the updated balance:

=>

import java.util.Scanner; public class ATMSystem {

public static void main(String[] args) { Scanner sc = new Scanner(System.*in*); double balance = 500.0;

int choice;

while (true) {

System.*out*.println("\n1. Check Balance\n2. Deposit\n3. Withdraw\n4. Exit"); System.*out*.print("Choose an option: ");

choice = sc.nextInt();

switch (choice) { case 1:

System.*out*.println("Balance: $" + balance); break;

case 2:

System.*out*.print("Enter deposit amount: $"); balance += sc.nextDouble(); System.*out*.println("Updated Balance: $" + balance); break;

case 3:

System.*out*.print("Enter withdrawal amount: $"); double withdraw = sc.nextDouble();

if (withdraw <= balance) { balance -= withdraw;

System.*out*.println("Updated Balance: $" + balance);

} else {

System.*out*.println("Insufficient balance!");

}

break; case 4:

System.*out*.println("Thank you! Goodbye!"); sc.close();

return; default:

System.*out*.println("Invalid choice!");

}

}

}

}

\*Unit-3:

1. In pune university Teacher wants to store the students whose name start with 'Y' alphabet create a software program for this. Which takes the input from user and if name start with 'Y' then print on console valid else invalid:

=>

import java.util.Scanner;

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

Scanner sc = new Scanner(System.in);

// Ask for student name System.out.print("Enter student's name: "); String name = sc.nextLine();

// Check if the name starts with 'Y'

if (name.length() > 0 && name.charAt(0) == 'Y') { System.out.println("Valid");

} else {

System.out.println("Invalid");

}

}

}

1. Create a basic text editor where the user can where the user can append and delete character from a string. Use stringbuffer class for this. Operation:

=>

import java.util.Scanner;

public class SimpleTextEditor {

public static void main(String[] args) { Scanner sc = new Scanner(System.in); StringBuffer text = new StringBuffer();

while (true) {

System.out.println("\nBasic Text Editor Options:"); System.out.println("1. Append Text"); System.out.println("2. Delete Text"); System.out.println("3. Show Text"); System.out.println("4. Exit"); System.out.print("Choose an option: ");

int choice = sc.nextInt();

sc.nextLine(); // Consume newline character

switch (choice) { case 1:

// Append Text

System.out.print("Enter text to append: "); String appendText = sc.nextLine(); text.append(appendText); System.out.println("Text after append: " + text); break;

case 2:

// Delete Text

System.out.print("Enter number of characters to delete: "); int lengthToDelete = scanner.nextInt();

if (lengthToDelete <= text.length() && lengthToDelete > 0) { text.delete(text.length() - lengthToDelete, text.length()); System.out.println("Text after deletion: " + text);

} else {

System.out.println("Invalid number of characters to delete.");

}

break;

case 3:

// Show Text

System.out.println("Current Text: " + text); break;

case 4:

// Exit

System.out.println("Exiting text editor. Goodbye!"); sc.close();

return;

default:

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

}

}

}

}

1. A publisher wants to ensure all sentences in a paragraph start with a capital letter. Write a program to capitalize the first letter of every sentence in a paragraph:

=>

import java.util.Scanner;

public class CapitalizeSentences {

public static void main(String[] args) { Scanner sc = new Scanner(System.in);

// Input paragraph System.out.print("Enter a paragraph: "); String paragraph = sc.nextLine();

// Split paragraph into sentences and capitalize the first letter String[] sentences = paragraph.split("(?<=\\.)"); StringBuilder result = new StringBuilder();

for (String sentence : sentences) { sentence = sentence.trim();

if (sentence.length() > 0) { result.append(Character.toUpperCase(sentence.charAt(0)))

.append(sentence.substring(1)).append(" ");

}

}

System.out.println("\nCapitalized Paragraph: " + result.toString().trim());

}

}

1. A cashier notices repeated items in the bill descriptions. Write a program to count the number of times each word appears in a string. By using the stringbuilder class:

=>

import java.util.Scanner; public class WordCount {

public static void main(String[] args) { Scanner scanner = new Scanner(System.in);

// Input bill description System.out.print("Enter the bill description: ");

String billDescription = scanner.nextLine().toLowerCase();

// Split the string into words

String[] words = billDescription.split("\\s+");

// Count and display occurrences StringBuilder result = new StringBuilder(); for (int i = 0; i < words.length; i++) {

int count = 1;

if (!result.toString().contains(words[i])) { for (int j = i + 1; j < words.length; j++) {

if (words[i].equals(words[j])) { count++;

words[j] = "";

}

}

result.append(words[i]).append(" : ").append(count).append("\n");

}

}

// Output the result

System.out.println("\nWord count:\n" + result);

}

}

1. A company has a list of employee names in a file, and they want to identify employees with atleast 6 characters in his name. Write a program to read the contents of a text filecontaining names and display only the names with more than 6 characters on the console:

=>

import java.io.BufferedReader; import java.io.FileReader; import java.io.IOException;

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

// Path to the file containing employee names String fileName = "employees.txt";

try (BufferedReader br = new BufferedReader(new FileReader(fileName))) { String name;

// Read each name from the file

while ((name = br.readLine()) != null) {

// Check if the name has more than 6 characters if (name.length() > 6) {

System.out.println(name);

}

}

} catch (IOException e) { e.printStackTrace();

}

}

}

\*How To Run:-

* 1. Create a file employees.txt with employee names. Example content: John

Samantha Michael Emily Alexandra Chris Jonathan

* 1. Save the Java file as EmployeeNameFilter.java.
  2. Compile and run the program.

\*This program filters and prints only the names with **more than 6 characters**.

1. Write a program to read names from an array reverse each name that ends with the letter 'a',and display the reversed names on the console:

=>

public class ReverseNames {

public static void main(String[] args) {

// Array of names

String[] names = {"Samantha", "John", "Maria", "Emily", "Anna", "Chris"};

// Loop through each name in the array for (String name : names) {

// Check if the name ends with 'a' if (name.endsWith("a")) {

// Reverse the name

String reversedName = new StringBuilder(name).reverse().toString(); System.out.println(reversedName);

}

}

}

}

\*How to Run:- Same as Previous.

\*Unit-4:

1. A cashier wants to record daily sales data (item name, quantity, price) in a text file. Create the program software for this and store the data into the file using byte streams:

=>

import java.io.FileOutputStream; import java.io.IOException; import java.util.Scanner;

public class SalesRecorder {

public static void main(String[] args) { Scanner sc = new Scanner(System.in); String fileName = "sales\_data.txt";

try (FileOutputStream fos = new FileOutputStream(fileName, true)) { while (true) {

System.out.print("Enter item name (or 'exit' to stop): "); String itemName = scanner.nextLine();

if (itemName.equalsIgnoreCase("exit")) break;

System.out.print("Enter quantity: ");

int quantity = Integer.parseInt(scanner.nextLine());

System.out.print("Enter price: ");

double price = Double.parseDouble(scanner.nextLine());

String salesData = "Item: " + itemName + ", Quantity: " + quantity + ", Price: " + price + "\n";

fos.write(salesData.getBytes());

}

} catch (IOException e) { e.printStackTrace();

} finally { scanner.close();

}

System.out.println("Sales recording ended.");

}

}

1. 2.A cashier’s daily sales data is stored in a file, including item names, quantities, and prices. How would you use serialization to save all the sales records into a file for future use?:

=>

import java.io.\*;

import java.util.Scanner;

class SaleRecord implements Serializable { String itemName;

int quantity; double price;

public SaleRecord(String itemName, int quantity, double price) { this.itemName = itemName;

this.quantity = quantity; this.price = price;

}

}

public class SalesRecorder {

public static void main(String[] args) { Scanner scanner = new Scanner(System.in);

try (ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream("sales\_data.ser"))) {

while (true) {

System.out.print("Enter item name (or 'exit' to stop): "); String itemName = scanner.nextLine();

if (itemName.equalsIgnoreCase("exit")) break; System.out.print("Enter quantity: ");

int quantity = Integer.parseInt(scanner.nextLine()); System.out.print("Enter price: ");

double price = Double.parseDouble(scanner.nextLine()); SaleRecord record = new SaleRecord(itemName, quantity, price); oos.writeObject(record);

}

} catch (IOException e) { e.printStackTrace();

} finally { scanner.close();

}

System.out.println("Sales recording ended.");

}

}

1. The cashier enters prices as part of the sales data. How can you use regex to validate that the price follows the format of a decimal number (e.g., "19.99")?write program ?:

=>

import java.util.Scanner; import java.util.regex.Pattern;

public class PriceValidator {

public static void main(String[] args) { Scanner scanner = new Scanner(System.in);

String pricePattern = "^[0-9]+(\\.[0-9]{2})?$";

while (true) {

System.out.print("Enter price (or 'exit' to stop): "); String price = scanner.nextLine();

if (price.equalsIgnoreCase("exit")) break; if (Pattern.matches(pricePattern, price)) { System.out.println("Price is valid.");

} else {

System.out.println("Invalid price format. Example: 19.99");

}

}

}

}

1. The program stores sales data for multiple items in a file. Each record occupies 50 bytes. If the cashier needs to modify the price of the second item in the file (starting from the beginning), how would RandomAccessFile be used to seek to the correct position and update the price while keeping the rest of the records intact:

=>

import java.io.\*;

public class ModifyPrice {

public static void main(String[] args) {

String fileName = "sales\_data.dat"; // Data file

try (RandomAccessFile file = new RandomAccessFile(fileName, "rw")) {

// Move to the second record (50 \* 1) file.seek(50 \* 1);

// Skip item name and quantity (40 bytes), then update the price (last 10 bytes) file.seek(file.getFilePointer() + 40);

// New price

String newPrice = "29.99";

byte[] priceBytes = newPrice.getBytes(); byte[] priceBytesToWrite = new byte[10];

System.arraycopy(priceBytes, 0, priceBytesToWrite, 0, Math.min(priceBytes.length, 10));

// Write new price file.write(priceBytesToWrite);

System.out.println("Updated Price: " + newPrice);

} catch (IOException e) { e.printStackTrace();

}

}

}

1. The cashier needs to store the sale date along with the item name and price in a text file. How would you store the Date object in a text file using byte streams:

=>

import java.io.\*; import java.util.Date;

class SaleRecord implements Serializable { String itemName;

double price; Date saleDate;

public SaleRecord(String itemName, double price, Date saleDate) { this.itemName = itemName;

this.price = price; this.saleDate = saleDate;

}

}

public class SaleDataStorage {

public static void main(String[] args) {

// Create a SaleRecord object with item, price, and current date

SaleRecord sale = new SaleRecord("Laptop", 899.99, new Date()); try (ObjectOutputStream oos = new ObjectOutputStream(new

FileOutputStream("sale\_data.dat"))) { oos.writeObject(sale); // Serialize and write to file

System.out.println("Sale record saved successfully.");

} catch (IOException e) { e.printStackTrace();

}

}

}

\*Unit-5

1. A company wants to create a user interface for customer feedback. The interface should include a text area where users can enter their feedback and a button to submit the feedback. Write a program using AWT to create this interface and handle the submission button click Event:

=>

import java.awt.\*; import java.awt.event.\*;

public class FeedbackForm extends Frame { TextArea feedbackArea;

Button submitButton; public FeedbackForm() {

setTitle("Customer Feedback"); setSize(400, 300); setLayout(new FlowLayout());

feedbackArea = new TextArea(10, 30); add(feedbackArea);

submitButton = new Button("Submit Feedback"); add(submitButton); submitButton.addActionListener(e -> {

String feedback = feedbackArea.getText();

System.out.println(feedback.isEmpty() ? "No feedback entered." : "Feedback Submitted: " + feedback);

});

addWindowListener(new WindowAdapter() { public void windowClosing(WindowEvent we) {

System.exit(0);

}

});

setVisible(true);

}

public static void main(String[] args) { new FeedbackForm();

}

}

1. A survey needs to be designed where users can select their gender (Male, Female, Other) from radio buttons. Write a Swing program to create a set of radio buttons for gender selection and display the selected option when a button is clicked:

=>

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

public class GenderSelectionSurvey extends JFrame { public GenderSelectionSurvey() {

setTitle("Gender Selection"); setSize(250, 150);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); setLayout(new FlowLayout());

// Radio buttons for gender selection

JRadioButton maleButton = new JRadioButton("Male"); JRadioButton femaleButton = new JRadioButton("Female"); JRadioButton otherButton = new JRadioButton("Other");

// Group the radio buttons

ButtonGroup group = new ButtonGroup(); group.add(maleButton); group.add(femaleButton); group.add(otherButton);

// Label to show selected gender

JLabel label = new JLabel("Select Gender:");

// Action listener to display selected gender

maleButton.addActionListener(e -> label.setText("Selected Gender: Male")); femaleButton.addActionListener(e -> label.setText("Selected Gender: Female")); otherButton.addActionListener(e -> label.setText("Selected Gender: Other"));

// Add components to frame add(maleButton); add(femaleButton); add(otherButton); add(label);

setVisible(true);

}

public static void main(String[] args) { new GenderSelectionSurvey();

}

}

1. Write a program that divides two numbers. Handle both ArithmeticException (for division by zero) and NumberFormatException (for invalid input) using multiple catch blocks:

=>

import java.util.Scanner;

public class DivisionExample {

public static void main(String[] args) { Scanner sc = new Scanner(System.in);

try {

// Input two numbers System.out.print("Enter numerator: ");

int num1 = Integer.parseInt(sc.nextLine()); System.out.print("Enter denominator: "); int num2 = Integer.parseInt(sc.nextLine());

// Division

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

} catch (ArithmeticException e) {

System.out.println("Error: Division by zero is not allowed.");

} catch (NumberFormatException e) {

System.out.println("Error: Invalid input, please enter valid integers.");

} finally { sc.close();

}

}

}

1. A company wants to create a user registration form using Java Swing with the following fields: A text field for the user's name (placed in the North region of the frame)

A text field for the user's email address (placed in the South region of the frame) A text field for the user's phone number (placed in the East region of the frame) A text field for the user's address (placed in the West region of the frame)

A "Submit" button (placed in the center of the frame) Write a program using Java Swing and BorderLayout:

=>

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

public class UserRegistrationForm extends JFrame { public UserRegistrationForm() {

// Set frame properties setTitle("User Registration"); setSize(400, 200);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); setLayout(new BorderLayout());

// Create text fields for user information

JTextField nameField = new JTextField("Enter your name"); JTextField emailField = new JTextField("Enter your email"); JTextField phoneField = new JTextField("Enter your phone number"); JTextField addressField = new JTextField("Enter your address");

// Create Submit button

JButton submitButton = new JButton("Submit");

// Add components to the frame using BorderLayout add(nameField, BorderLayout.NORTH); add(emailField, BorderLayout.SOUTH); add(phoneField, BorderLayout.EAST); add(addressField, BorderLayout.WEST);

add(submitButton, BorderLayout.CENTER);

// Make the frame visible setVisible(true);

}

public static void main(String[] args) { new UserRegistrationForm();

}

}

1. Write java program that takes an integer input from the user if the user enters a non-integer value, catch the NumberFormatException and print the message “please enter the valid number:

=>

import java.util.Scanner; public class IntegerInput {

public static void main(String[] args) { Scanner sc = new Scanner(System.in);

try {

System.out.print("Enter an integer: ");

int number = Integer.parseInt(sc.nextLine()); System.out.println("You entered: " + number);

} catch (NumberFormatException e) { System.out.println("Please enter a valid number.");

} finally { sc.close();

}

}

}