Section1:

------------

1. Write a Java program that takes an integer input from the user and checks if it is even or odd.

Code:

import java.util.Scanner;

public class Evenodd{

public static void main(String[] args){

Scanner sc=new Scanner(System.in);

System.out.print("Enetr no");

int no=sc.nextInt();

if(no%2==0){

System.out.print(no + "is even");

}

else{

System.out.print(no + "is odd");

}

sc.close();

}

}

2) Implement a simple calculator that performs addition, subtraction, multiplication, and division based on user input.

Code 2:

import java.util.Scanner;

public class calc {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter first no: ");

double num1 = scanner.nextDouble();

System.out.print("Enter second no: ");

double num2 = scanner.nextDouble();

System.out.print("Enter an operator +, -, \*, / ");

char operator = scanner.next().charAt(0);

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:

System.out.println("Invalid operator!");

return;

}

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

scanner.close();

}

}

Section 2:

-------------

2)1) Create a Student class with fields like name, roll number, and marks. Add methods to display details and calculate grade.

Code:

public class Student {

String name;

int rollno;

double marks;

public Student(String name, int rollno, double marks) {

this.name = name;

this.rollno = rollno;

this.marks = marks;

}

void displayDetails() {

System.out.println("Name: " + name);

System.out.println("Roll Number: " + rollno);

System.out.println("Marks: " + marks);

}

String calcgrade() {

if (marks >= 90) {

return "A";

} else if (marks >= 80) {

return "B";

} else if (marks >= 70) {

return "C";

} else if (marks >= 60) {

return "D";

} else {

return "F";

}

}

public static void main(String[] args) {

Student stu = new Student("Abhinav", 11, 90);

stu.displayDetails();

System.out.println("Grade: " + stu.calcgrade());

}

}

2)2) Create a class BankAccount with methods to deposit, withdraw, and display balance. Apply basic validation (e.g., can't withdraw more than balance).

Code:

public class BankAccount {

private double balance;

public void deposit(double amount) {

if (amount > 0) {

balance= balance+amount;

System.out.println("Deposited: " + amount);

} else {

System.out.println("Invalid deposit amount.");

}

}

public void withdraw(double amount) {

if (amount > 0 && amount <= balance) {

balance=balance-amount;

System.out.println("Withdraw: " + amount);

} else if (amount > balance) {

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

} else {

System.out.println("Invalid withdrawal amount.");

}

}

public void displayBalance() {

System.out.println("Current balance: " + balance);

}

public static void main(String[] args) {

BankAccount account = new BankAccount();

account.deposit(1000);

account.displayBalance();

account.withdraw(500);

account.displayBalance();

account.withdraw(600);

account.displayBalance();

}

}

Section 3:

----------------

3)1) Write a Java program to reverse the elements of an integer array without using a built-in function.

Code:

import java.util.Scanner;

public class ReverseArray {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int size = scanner.nextInt();

int[] array = new int[size];

System.out.println("Enter the elements of the array:");

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

array[i] = scanner.nextInt();

}

reverseArray(array);

System.out.println("Reversed array:");

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

System.out.print(array[i] + " ");

}

scanner.close();

}

public static void reverseArray(int[] array) {

int start = 0;

int end = array.length - 1;

while (start < end) {

int temp = array[start];

array[start] = array[end];

array[end] = temp;

start++;

end--;

}

}

}

3)2) Sort Array in Ascending Order

Code:

import java.util.Scanner;

public class SortArray {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int size = scanner.nextInt();

int[] array = new int[size];

System.out.println("Enter the elements of the array:");

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

array[i] = scanner.nextInt();

}

sortArray(array);

System.out.println("Sorted array in ascending order:");

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

System.out.print(array[i] + " ");

}

scanner.close();

}

public static void sortArray(int[] array) {

int n = array.length;

for (int i = 0; i < n - 1; i++) {

for (int j = 0; j < n - i - 1; j++) {

if (array[j] > array[j + 1]) {

int temp = array[j];

array[j] = array[j + 1];

array[j + 1] = temp;}}}}}

3)3) Identify and print all duplicates in an array.

Code:

import java.util.HashMap;

public class Charfreq {

public static void main(String[] args) {

String a = "142522";

HashMap<Character, Integer> map = new HashMap<Character, Integer>();

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

char currentChar = a.charAt(i);

if (map.containsKey(currentChar)) {

int n = map.get(currentChar) + 1;

map.put(currentChar, n);

} else {

map.put(currentChar, 1);

}

}

System.out.println(map);

}

}

3)4) Frequency Count Using HashMap

Code:

import java.util.HashMap;

public class Charfreq {

public static void main(String[] args) {

String a = "Abhinav";

HashMap<Character, Integer> map = new HashMap<Character, Integer>();

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

char currentChar = a.charAt(i);

if (map.containsKey(currentChar)) {

int n = map.get(currentChar) + 1;

map.put(currentChar, n);

} else {

map.put(currentChar, 1);

}

}

System.out.println(map);

}

}