ASSIGNMENT-1

Name:-ch.Mahanvitha

Reg.No:-19221785

Course code:-CSA0914

Course Name:-
Programming in Java
for Raspberry Pi

Faculty Name:-
Dr.Hemavathi R.

nario1 : Student Grading System.

nario: A teacher needs a simple program to lculate students grades based on their cores. The grading criteria are as follows:

- Score >= 90 : Grade A
- Score >= 80 and <90 : Grade B
- Score >= 70 and <80 : Grade C
- Score >= 60 and <70 : Grade D.
- Score < 60 : Grade F.

Question :

1) Design a Java Program that takes a student's score as input and outputs the corresponding grade using an if-else control structure.

2) Test case: • Input: 85
• Expected output: Grade B.

3) Additional Requirements: Implement a loop that allows the teacher to enter scores for multiple students and display the grades for each one until the teacher decides to stop.

Aim:- To write a java program for student Grading system that calculates students grades based on their scores.

Pseudo code:- 1) Start

2) Initialize a loop to continue until user decides to stop.

3) Enter the student's score.

4) Check score and assign the grade:

- Score >=90: Grade A
- Score >=80 and <90: Grade B
- Score >=70 and <80 : Grade C
- Score >=60 and <70: Grade D
- Score <60 : Grade F.

5) Display output.

Program :-

```
import java.util.Scanner;
public class StudentGradingSystem {
    public static void main(String[] args) {
        Scanner scanner=new Scanner(System.in);
        String continueGrading = "yes";
      while (continueGrading.equalsIgnoreCase("yes"))
      {
    System.out.print("Enter the student's
                                Score: ");
    int score = scanner.nextInt();
    char grade;
    if (score >=90) {
            grade = 'A';
    } else if (score >= 80){
            grade = 'B';
    }else if (score >=70){
             grade = 'C';
    }else if (score >=60){
             grade = 'D';
```

```
else {
    grade = 'F';
}
System.out.println("Grade: "+grade);
System.out.print("Do you want to enter
        another score? (y/n): ");
continueGrading = scanner.next();
}
System.out.println("Grading process finished");
}
}
```

Input:- Enter the Student's Score: 85

output:- Grade: Grade B.

2) Scenario2: Number Guessing Game.

Scenario: A simple number guessing game where the program randomly selects a number b/w 1 and 10, and the player has to guess it. the player has 3 attempts to guess number.

Question: 1) Implement a Java program that generates a random number b/w 1 and 10. Use for loop to give player 3 attempts to guess number. After each incorrect guess, the program provide a hit ("Too high" or "Too low").

2) Test case: 1) Random number: 7
2) player inputs: 5, 8, 7.

3) Expected O/p: "Too Low", "Too high", "correct"
3) Additional Requirement: If player fails to guess the number in three attempts, the program should reveal the correct number and offer to play again using while loop.

Aim:- To write a java program that allows the player to guess a random number between 1 and 10.

Pseudo Code:- 1) Start
2) Initialize the game loop (while loop) to allow replaying.
3) Generate a number between 1 and 10
4) player has three attempts to guess.
5) If correct, display success
If incorrect, provide hint ("Too high" or "Too low").
6) End the program.

Program:-

```
import java.util.Random;
import java.util.Scanner;
public class NumberGuessingGame {
    public static void main(String[] args){
        Scanner scanner=new Scanner(System.in);
        Random random = new Random();
        String playAgain = "yes";
        while(playAgain.equalsIgnoreCase("yes")) {
```

```
    numberToGuess = random.nextInt(10)+1;
boolean guessedCorrectly = false;
System.out.println("I have selected a number
between 1 and 10. You have 3 attempts to
guess it");
for (int i=1; i<=3; i++) {
    System.out.print("Attempt "+i+": Enter
                                your guess :");
  int playerGuess = scanner.nextInt();
 if (playerGuess == numberToGuess) {
    System.out.println("Correct! You guessed
                it in "+i+"attempts.");
    guessedCorrectly = true;
    break;
  } else if (playerGuess < numberToGuess) {
      System.out.println("Too low.");
  }else {
      System.out.println("Too high.");
  }
}
if (!guessedCorrectly){
    System.out.println("Sorry, you've used
all your attems. The correct number was"
+numberToGuess+".");
}
System.out.print("Do you want to play again?
                        (y/n):");
```

```
        playAgain = scanner.next();
    }
    System.out.println("Thanks for playing!");
  }
}
```

Input;- Attempt 1:
output Enter your guess: 5
Too high:
Attempt 2: Enter your guess: 8
Too high
Attempt 3: Enter your guess: 7
Too high
Sorry, you've used all your attempts. The correct answer is 1.

3) Scenario 3: Multiplication Table Generator.
Scenario: A School requires a program to generate and display the multiplication table for any number entered by the user.
Question: 1) Create a Java program that takes a number as input and uses a for loop to generate and print the multiplication table.
2) Test case: 1) Input: 5
2) Expected output:

5 X 1 = 5
5 X 2 = 10
5 X 3 = 15
- - - -
5 X 10 = 50.

ditional Requirement: Modify the program; allow the user to specify the range of ne multiplication table (e.g. 1 to 15).

Aim:- To write a java program that generates and displays multiplication table.

Pseudo code:-

1) Start.
2) enter a number.
3) use for loop to iterate from 1 to the specified range.
4) Multiply entered number by loop.
5) Display output.

Program:-

```
import java.util.Scanner;
public class MultiplicationTable{
    public static void main(String[] args){
        Scanner scanner=new Scanner(System.in);
        System.out.print("Enter number: ");
        int number=scanner.nextInt();
        System.out.print("Enter range : ");
        int range=scanner.nextInt();
        System.out.println("Multiplication Table
                for "+number+"up to "+range+
                                ":");
        for(int i=1; i<=range; i++){
```

```
System.out.println(number + "x" + i + "=" +
                   (number * i));
    }
  }
}
```

Input :- Enter number : 5
Enter range : 10

output :- Multiplication table:

5 X 1 = 5
5 X 2 = 10
5 X 3 = 15
5 X 4 = 20
5 X 5 = 25
5 X 6 = 30
5 X 7 = 35
5 X 8 = 40
5 X 9 = 45
5 X 10 = 50

4) Scenario 4: Even and odd Number Counter.
Scenario: A simple utility program is needed to count how many even and odd numbers are in a given list of integers.
Question: i) Develop a Java program that takes an array of integers as input and uses a for loop to count how many even and odd numbers are in the array. Use an if-else statement to determine if a number is even or odd.

st case: 1) Input : {2,3,4,5,6}
2) Expected o/p: evencount:3, oddcount:2.

Additional Requirement: Calculate sum of even numbers and odd numbers.

Aim:- To write a java program that counts how many even and odd numbers are in a given array of integers. calculate sum of even and odd numbers.

Pseudo code:-

1) Start.
2) Initialize counters for even and odd numbers.
3) Initialize sum variables for even and odd numbers.
4) Loop through the array of integers.
5) If even, increment even counter and add number to even sum.
6) If odd, increment odd counter and add number to odd sum.
7) Print sum of even, odd numbers.
8) Display output.

program:-

```
import java.util.Scanner;

public class EvenOddCounter {
    public static void main(String[] args){
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter number of
                          elements:");
```

```
int n= scanner.nextInt();
int [] numbers = new int [n];
System.out.println("Enter elements of array:");
for(int i=0; i<n; i++){
    numbers [i] = scanner.nextInt();
}
int evenCount =0, oddCount =0;
int evenSum=0, oddSum=0;
for(int num :numbers){
    if(num%2==0){
        evenCount++;
        evenSum += num;
    }else{
        oddCount++;
        oddSum+=num;
    }
}
System.out.println("Even count: "+evenCount+
                   ", Odd count: "+oddCount);
System.out.println("Sum of even numbers:"
                   +evenSum+"; Sum of
                   odd numbers: "+oddSum);
}
}
```

Input:- Enter number of elements: 3
Enter elements of array: 25 10 46

output:-

Even count: 2 , Odd count: 1

Sum of even numbers: 56, Sum of odd numbers: 25

nario 5: Simple ATM Simulation.
enario: Simulate a basic ATM System where
e user can choose from three options:
check balance, deposit money, or withdraw
money. The initial balance is set to $1000.
Question: 1) Write a java program that pres-
-ents a menu to the user using a switch
statement. Based on the user's selection, the
program should perform the appropriate
action: check balance, deposit money, or
withdraw money. Use a loop to allow the
user to perform multiple actions until they
choose to exit.
2) Test case: • Initial Balance: $1000.
• User Actions: Deposit $200, Withdraw $150,
Check Balance.
• Expected output: Balance: $1050.
3) Additional Requirement: Ensure that the pro-
-gram prevents the user from withdrawing
more money than the current balance
by displaying an error message.

Aim:- To write a java program that simulates
a ATM System.

Pseudo code:- 1) Start.
2) Set initial balance to $1000.
3) Use loop to display menu: check balance,
deposit money, withdraw money, and exit.
4) Use switch statement to check
balance, deposit, withdrawl, and exit.

5) Display output.

Program:-

```
import java.util.Scanner;
Public class SimpleATMSimulation{
    Public Static void main(String[] args){
        Scanner scanner=new Scanner(System.in);
        int balance = 1000;
        boolean exit = false;
       while (!exit){
          System.out.println("\nATM menu:");
        System.out.println("1. check Balance");
        System.out.println("2. Deposit money");
        System.out.println("3. withdraw money");
        System.out.println("4. Exit");
       System.out.println("choose option:");
       int choice = scanner.nextInt();
       switch (choice) {
          case 1:
            System.out.println("Current Balance:
                                $"+balance);
            break;
          case 2:
            System.out.println("Enter
                deposit amount :$");
          int deposit =scanner.nextInt();
          balance += deposit;
```

```
System.out.println("Updated Balance: $" +
                                    balance);
break;
case 3:
System.out.print("Enter withdrawl amount:
                                        $");
int withdrawl = scanner.nextInt();
if (withdrawl = balance) {
  System.out.println("Error: Insufficient
                                    funds.");
}else{
   balance -= withdrawl;
System.out.println("Updated Balance: $" + balance);
  }
  break;
case 4:
   exit = true;
  System.out.println("Exiting ATM. Have a
nice day!");
  break;
  default:
  System.out.println("Invalid option. Please
                                    try again.");

  break;
}}}}
```

Input:- ATM Menu: 1. check Balance
2. Deposit Money
3. withdrawl Money
4. Exit.
Choose option: 2

Output:- Enter deposit amount : $500
Updated Balance : $1500.