TASK 1

1. Generate a random number within a specified range, such as 1 to 100.

2. Prompt the user to enter their guess for the generated number.

3. Compare the user's guess with the generated number and provide feedback on whether the guess

is correct, too high, or too low.

4. Repeat steps 2 and 3 until the user guesses the correct number.

You can incorporate additional details as follows:

5. Limit the number of attempts the user has to guess the number.

6. Add the option for multiple rounds, allowing the user to play again.

7. Display the user's score, which can be based on the number of attempts taken or rounds won.

Code:

import java.util.Random;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

Random random = new Random();

int min = 1;

int max = 100;

int attempts = 0;

int score = 0;

boolean playAgain = true;

System.out.println("Welcome to the Number Guessing Game!");

while (playAgain) {

int randomNumber = random.nextInt(max - min + 1) + min;

int remainingAttempts = 10; // Limit the number of attempts

System.out.println("I've selected a random number between " + min + " and " + max + ". Try to guess it!");

while (remainingAttempts > 0) {

System.out.print("Enter your guess (remaining attempts: " + remainingAttempts + "): ");

int userGuess = scanner.nextInt();

attempts++;

remainingAttempts--;

if (userGuess == randomNumber) {

System.out.println("Congratulations! You guessed the correct number: " + randomNumber);

score++;

break;

} else if (userGuess < randomNumber) {

System.out.println("Your guess is too low.");

} else {

System.out.println("Your guess is too high.");

}

if (remainingAttempts == 0) {

System.out.println("You've run out of attempts. The correct number was: " + randomNumber);

}

}

System.out.print("Do you want to play again? (yes/no): ");

String playAgainResponse = scanner.next().toLowerCase();

if (playAgainResponse.equals("no")) {

playAgain = false;

}

}

System.out.println("Game over! You played " + attempts + " times and won " + score + " rounds.");

scanner.close();

}

}

TASK 2

Input: Take marks obtained (out of 100) in each subject.

Calculate Total Marks: Sum up the marks obtained in all subjects.

Calculate Average Percentage: Divide the total marks by the total number of subjects to get the

average percentage.

Grade Calculation: Assign grades based on the average percentage achieved.

Display Results: Show the total marks, average percentage, and the corresponding grade to the user

Program:

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Initialize variables

int totalMarks = 0;

int totalSubjects;

double averagePercentage;

// Input the total number of subjects

System.out.print("Enter the total number of subjects: ");

totalSubjects = scanner.nextInt();

// Input marks for each subject and calculate total marks

for (int i = 1; i <= totalSubjects; i++) {

System.out.print("Enter marks for subject " + i + ": ");

int marks = scanner.nextInt();

totalMarks += marks;

}

// Calculate average percentage

averagePercentage = (double) totalMarks / totalSubjects;

// Assign grades based on the average percentage

char grade;

if (averagePercentage >= 90) {

grade = 'A';

} else if (averagePercentage >= 80) {

grade = 'B';

} else if (averagePercentage >= 70) {

grade = 'C';

} else if (averagePercentage >= 60) {

grade = 'D';

} else {

grade = 'F';

}

// Display results

System.out.println("Total Marks: " + totalMarks);

System.out.println("Average Percentage: " + averagePercentage + "%");

System.out.println("Grade: " + grade);

// Close the scanner

scanner.close();

}

}

TASK 3

1.Create a class to represent the ATM machine.

2. Design the user interface for the ATM, including options such as withdrawing, depositing, and

checking the balance.

3. Implement methods for each option, such as withdraw(amount), deposit(amount), and

checkBalance().

4. Create a class to represent the user's bank account, which stores the account balance.

5. Connect the ATM class with the user's bank account class to access and modify the account

balance.

6. Validate user input to ensure it is within acceptable limits (e.g., sufficient balance for withdrawals).

7. Display appropriate messages to the user based on their chosen options and the success or failure

of their transactions.

ATM INTERFACE

Program:

import java.util.Scanner;

class BankAccount {

private double balance;

public BankAccount(double initialBalance) {

this.balance = initialBalance;

}

public double getBalance() {

return balance;

}

public void deposit(double amount) {

if (amount > 0) {

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 -= amount;

System.out.println("Withdrawn $" + amount);

} else {

System.out.println("Invalid withdrawal amount or insufficient balance.");

}

}

}

class ATM {

private BankAccount account;

public ATM(BankAccount account) {

this.account = account;

}

public void displayMenu() {

System.out.println("Welcome to the ATM!");

System.out.println("1. Check Balance");

System.out.println("2. Deposit");

System.out.println("3. Withdraw");

System.out.println("4. Exit");

}

public void start() {

Scanner scanner = new Scanner(System.in);

while (true) {

displayMenu();

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

int choice = scanner.nextInt();

switch (choice) {

case 1:

System.out.println("Balance: $" + account.getBalance());

break;

case 2:

System.out.print("Enter deposit amount: $");

double depositAmount = scanner.nextDouble();

account.deposit(depositAmount);

break;

case 3:

System.out.print("Enter withdrawal amount: $");

double withdrawAmount = scanner.nextDouble();

account.withdraw(withdrawAmount);

break;

case 4:

System.out.println("Thank you for using the ATM!");

scanner.close();

return;

default:

System.out.println("Invalid choice. Please select a valid option.");

}

}

}

}

public class Main {

public static void main(String[] args) {

// Create a bank account with an initial balance of $1000

BankAccount userAccount = new BankAccount(1000);

// Create an ATM instance connected to the user's account

ATM atm = new ATM(userAccount);

// Start the ATM

atm.start();

}

}