Task 1

Number Game

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

```java
package internship;
import java.util.*;
public class Demo
{
public static void main(String args[])
{
int minRange=1;
int maxRange=100;
int randomNum=getRandomNumber(minRange,maxRange);
System.out.println("Random Number:"+randomNum);
}
private static int getRandomNumber(int min,int max)
{
Random rand=new Random();
return rand.nextInt((max-min)+1)+min;
}
}
```

```
Console
<terminated> Demo (1) [Java Application] C:\Program Files\Java\jdk-20\bin\javaw.exe  (30-Mar-2024, 3:00:48 pm – 3:00:50 pm) [pid:
Random Number:10
```

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

```java
package internship;

import java.util.Scanner;
public class GuessNumber{
public static void main(String args[])
{
int randomNumber=(int)(Math.random()*100)+1;
Scanner sc=new Scanner(System.in);
System.out.println("Guess the number between 1and 100");
int userGuess=sc.nextInt();
if(userGuess==randomNumber){
System.out.println("Congratulation you guess correct number.");
}
else{
System.out.println("Sorry the correct number was:"+randomNumber);
}
sc.close();
}
}
```

```
Console ×
<terminated> GuessNumber [Java Application] C:\Program Files\Java\jdk-20\bin\javaw.exe  (30-Mar-2024, 3:07:39 pm – 3:07:41 pm) [pid: 6312]
Guess the number between 1and 100
8
Sorry the correct number was:41
```

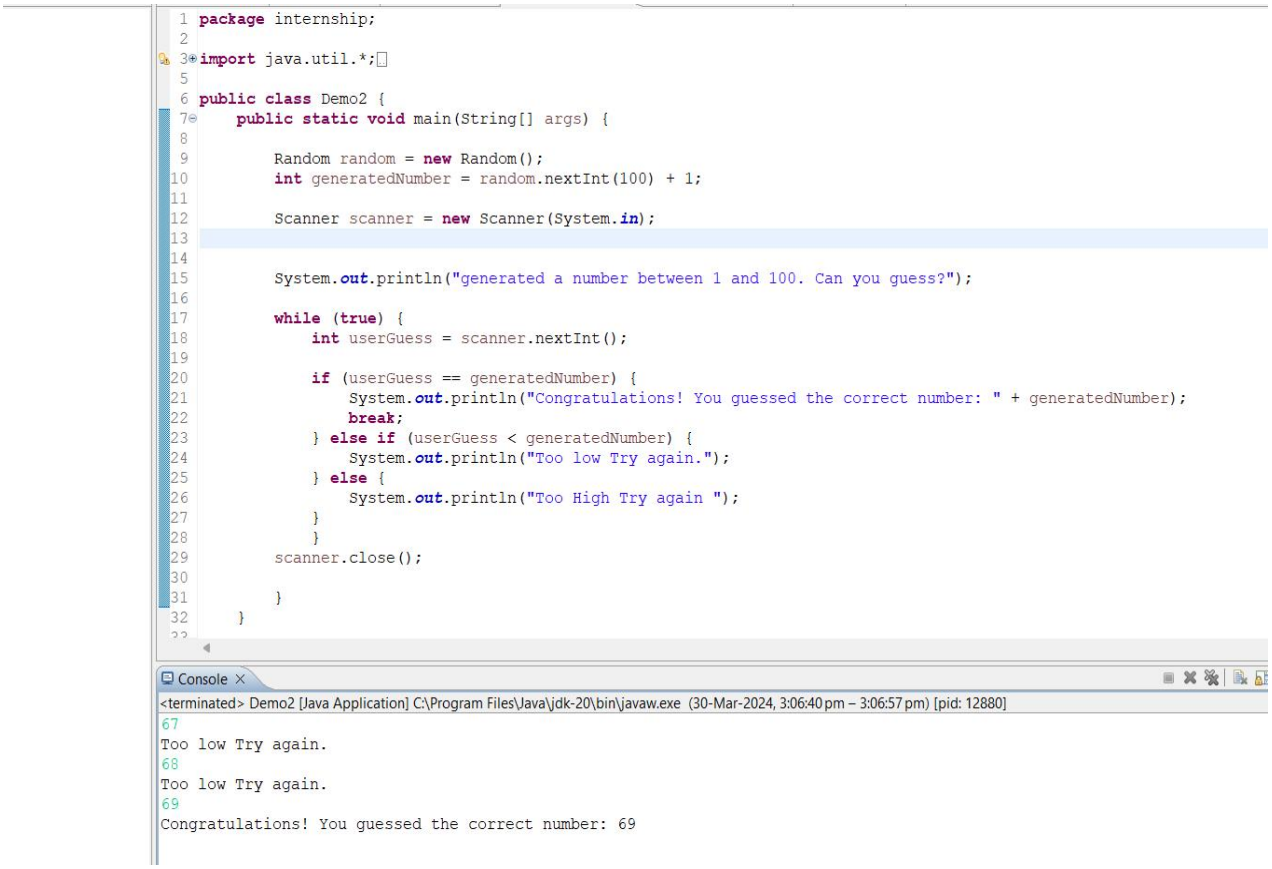
Compare the user's guess with the generated number and provide feedback on whether the ques is correct, too high, or too low.

```java
package internship;

import java.util.Scanner;

public class Demo1 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        int minRange = 1;
        int maxRange = 100;
        int randomNumber = (int) (Math.random() * (maxRange - minRange + 1)) + minRange;
        int userGuess;

        System.out.println("Welcome to Guess the Number Game!");
        System.out.println("I have selected a number between " + minRange + " and " + maxRange );

        while (true) {

            System.out.print("Enter your guess: ");
            userGuess = scanner.nextInt();
            if (userGuess < randomNumber) {
                System.out.println("Too low! Try again.");
            } else if (userGuess > randomNumber) {
                System.out.println("Too high! Try again.");
            } else {
                System.out.println("Congratulations! You guessed the correct number: " + randomNumber);
                break;
            }
        }

        scanner.close();
    }
```

Console

```
<terminated> Demo1 [Java Application] C:\Program Files\Java\jdk-20\bin\javaw.exe  (30-Mar-2024, 3:02:24 pm – 3:02:50 pm) [pid: 19064]
Enter your guess: 22
Too high! Try again.
Enter your guess: 21
Too high! Try again.
Enter your guess: 12
Congratulations! You guessed the correct number: 12
```

- Repeat steps 2 and 3 until the user guesses the correct number.
- You can incorporate additional details as follows:
- Limit the number of attempts the user has to guess the number.
- Add the option for multiple rounds, allowing the user to play again.
- Display the user's score, which can be based on the number of attempts taken or rounds won.



```java
package internship;

import java.util.*;

public class Demo2 {
    public static void main(String[] args) {

        Random random = new Random();
        int generatedNumber = random.nextInt(100) + 1;

        Scanner scanner = new Scanner(System.in);


        System.out.println("generated a number between 1 and 100. Can you guess?");

        while (true) {
            int userGuess = scanner.nextInt();

            if (userGuess == generatedNumber) {
                System.out.println("Congratulations! You guessed the correct number: " + generatedNumber);
                break;
            } else if (userGuess < generatedNumber) {
                System.out.println("Too low Try again.");
            } else {
                System.out.println("Too High Try again ");
            }
            }
        scanner.close();

        }
    }
```

```
Console
<terminated> Demo2 [Java Application] C:\Program Files\Java\jdk-20\bin\javaw.exe  (30-Mar-2024, 3:06:40 pm – 3:06:57 pm) [pid: 12880]
67
Too low Try again.
68
Too low Try again.
69
Congratulations! You guessed the correct number: 69
```

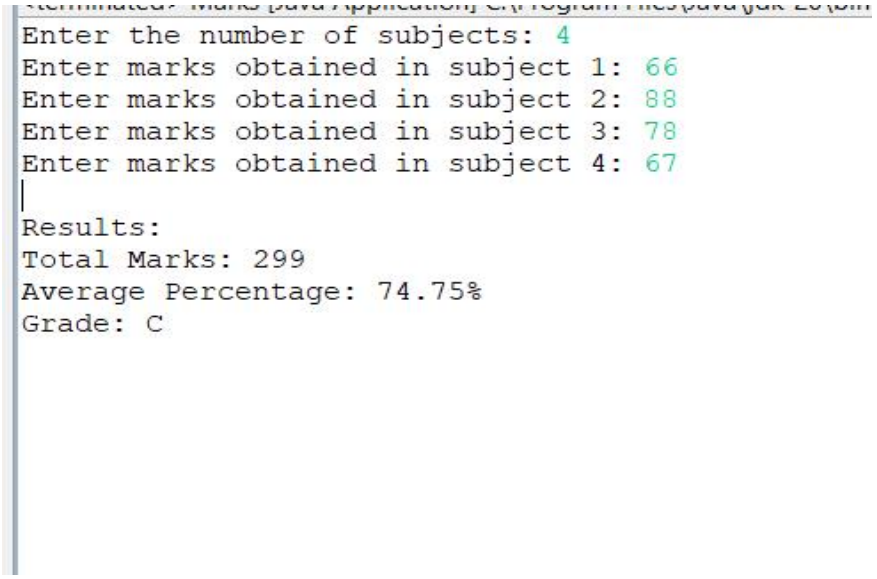
Task 2

STUDENT GRADE CALCULATOR

- 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..

```java
package task2;
import java.util.Scanner;

public class Marks {

	public static void main(String[] args) {
		Scanner scanner = new Scanner(System.in);
		System.out.print("Enter the number of subjects: ");
		int sub = scanner.nextInt();

		int totalmarks = 0;
		int marks;

		for (int i = 1; i <= sub; i++) {
			System.out.print("Enter marks obtained in subject " + i + ": ");
			marks = scanner.nextInt();
			totalmarks =totalmarks+marks;
		}

		double averagePercentage = (double) totalmarks / sub;

		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';
		}

		System.out.println("\nResults:");
		System.out.println("Total Marks: " + totalmarks);
		System.out.println("Average Percentage: " + averagePercentage + "%");
		System.out.println("Grade: " + grade);
		scanner.close();

 }
}
```

Output:


Enter the number of subjects: 4
Enter marks obtained in subject 1: 66
Enter marks obtained in subject 2: 88
Enter marks obtained in subject 3: 78
Enter marks obtained in subject 4: 67
Results:
Total Marks: 299
Average Percentage: 74.75%
Grade: C

TASK 3

ATM INTERFACE

- Create a class to represent the ATM machine.
- Design the user interface for the ATM, including options such as withdrawing, depositing, and checking the balance.
- Implement methods for each option, such as withdraw(amount) deposit(amount) and checkBalance().
- Create a class to represent the user's bank account, which stores the account balance.
- Connect the ATM class with the user's bank account class to access and modify the account balance.
- Validate user input to ensure it is within acceptable limits (eg. sufficient balance for withdrawals)
- Display appropriate messages to the user based on their chosen options and the success or failure of their transactions.

```java
package task3;
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) {
     balance += amount;
     System.out.println("Deposit successful. Current balance: " + balance);
 }

 public boolean withdraw(double amount) {
     if (amount <= balance) {
         balance -= amount;
         System.out.println("Withdrawal successful. Current balance: " + balance);
         return true;
     } else {
         System.out.println("Insufficient funds. Withdrawal failed.");
         return false;
     }
 }
}


class ATMMachine {
 private BankAccount bankAccount;

 public ATMMachine(BankAccount bankAccount)
 {
     this.bankAccount = bankAccount;
 }
```

```
public void displayMenu() {
    System.out.println("Welcome to the ATM Machine");
    System.out.println("1. Withdraw");
    System.out.println("2. Deposit");
    System.out.println("3. Check Balance");
    System.out.println("4. Exit");
}

public void withdraw(double amount) {
    bankAccount.withdraw(amount);
}

public void deposit(double amount) {
    bankAccount.deposit(amount);
}

public void checkBalance() {
    System.out.println("Current balance: " + bankAccount.getBalance());
}
}

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

    BankAccount userAccount = new BankAccount(1000);


    ATMMachine atm = new ATMMachine(userAccount);

    int choice;
    do {
        // Display menu
        atm.displayMenu();
        System.out.print("Enter your choice: ");
        choice = scanner.nextInt();
```

```
        int choice;
        do {
            // Display menu
            atm.displayMenu();
            System.out.print("Enter your choice: ");
            choice = scanner.nextInt();

            switch (choice) {
                case 1:
                    System.out.print("Enter amount to withdraw: ");
                    double withdrawAmount = scanner.nextDouble();
                    atm.withdraw(withdrawAmount);
                    break;
                case 2:
                    System.out.print("Enter amount to deposit: ");
                    double depositAmount = scanner.nextDouble();
                    atm.deposit(depositAmount);
                    break;
                case 3:
                    atm.checkBalance();
                    break;
                case 4:
                    System.out.println("Exiting ATM. Thank you!");
                    break;
                default:
                    System.out.println("Invalid choice. Please try again.");
                    break;
            }
        } while (choice != 4);
        scanner.close();

    }
}
```

Output:

```
Welcome to the ATM Machine
1. Withdraw
2. Deposit
3. Check Balance
4. Exit
Enter your choice: 1
Enter amount to withdraw: 400
Withdrawal successful. Current balance: 600.0
Welcome to the ATM Machine
1. Withdraw
2. Deposit
3. Check Balance
4. Exit
Enter your choice: 2
Enter amount to deposit: 10000
Deposit successful. Current balance: 10600.0
Welcome to the ATM Machine
1. Withdraw
2. Deposit
3. Check Balance
4. Exit
Enter your choice: 3
Current balance: 10600.0
Welcome to the ATM Machine
1. Withdraw
2. Deposit
3. Check Balance
4. Exit
Enter your choice:
```