

## TABLE OF CONTENTS

<b>QUESTION ONE</b> .....	2
<b>CREATE A JAVA PROGRAM THAT WILL COUNT ALL WORDS IN A SENTENCE. THE PROGRAM SHOULD HAVE A MINIMUM OF TWO CLASSES. (40)</b> .....	2
<b>1.1) Requirements for the first class:</b> .....	2
<b>1.2) Requirements for the second class:</b> .....	2
<b>Below is the entire code to execute the program:</b> .....	2
<b>Below is the output of the code after collecting the data from the user:</b> .....	3
<b>1.3) Construct a flowchart for class1 and class2 both combined. (10)</b> .....	4
<b>QUESTION TWO</b> .....	5
<b>CREATE A JAVA PROGRAM THAT WILL DISPLAY THE FIRST 40 PENTAGONAL NUMBERS. (20)</b> .....	5
<b>Below is the entire code to execute the program:</b> .....	5
<b>Below is the output of the above code:</b> .....	5
<b>QUESTION THREE</b> .....	6
<b>WRITE A JAVA PROGRAM THAT WILL COMPUTE THE FUTURE INVESTMENT VALUE AT A GIVEN INTEREST RATE FOR A SPECIFIED NUMBER OF YEARS. THE JAVA PROGRAM SHOULD HAVE A MINIMUM OF TWO CLASSES. (40)</b> .....	6
<b>3.1) Requirements for the first class:</b> .....	6
<b>3.2) Requirements for the second class:</b> .....	6
<b>Below is the entire code to execute the program:</b> .....	6
<b>Below is the output of the code after gathering data from the user:</b> .....	7
<b>REFERENCES</b> .....	8

## QUESTION ONE

**CREATE A JAVA PROGRAM THAT WILL COUNT ALL WORDS IN A SENTENCE. THE PROGRAM SHOULD HAVE A MINIMUM OF TWO CLASSES. (40)**

### **1.1) Requirements for the first class:**

- ❖ The first class should be named class1.
- ❖ The main method (starting point). **(5)**
- ❖ The object of class2. **(5)**
- ❖ It should also call the get method count\_words(String str). **(5)**

### **1.2) Requirements for the second class:**

- ❖ The second class should be named class2.
- ❖ It should have a constructor. **(5)**
- ❖ A get method named count\_words(String str). **(10)**

**Below is the entire code to execute the program:**

```
import java.util.Scanner;
import java.util.StringTokenizer;

class Class1{
    public static void main(String[] args){

        Scanner inputKeyBoard = new Scanner(System.in);
        System.out.print("Please enter your sentence: ");
        String userInput = inputKeyBoard.nextLine();
        System.out.println("Output: " + userInput);

        Class2 object = new Class2();
        int totalWords = object.count_words(userInput);

        System.out.println("There are " + totalWords + " words in the sentence inputted.");
    }
}

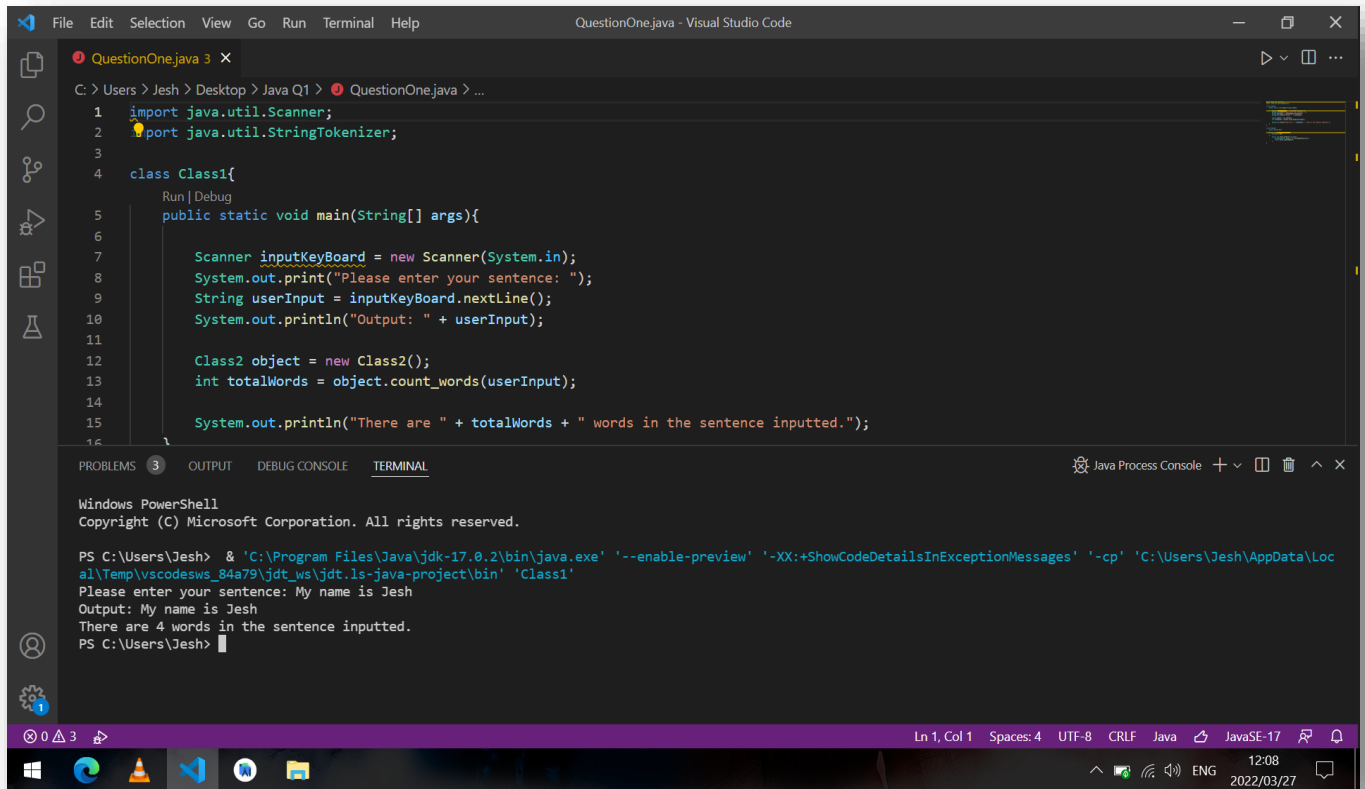
class Class2{
    public String str;

    public void Class2(String str){
        this.str = str;
    }

    public int count_words(String str){
        StringTokenizer words = new StringTokenizer(str);
        return words.countTokens();
    }
}
```

```
}  
}
```

***Below is the output of the code after collecting the data from the user:***



The screenshot displays the Visual Studio Code editor with a Java file named `QuestionOne.java`. The code defines a `Class1` with a `main` method that uses `Scanner` to read user input and `Class2` to count words. The terminal window shows the command to run the program, the user input "My name is Jesh", and the program's output: "Please enter your sentence: ", "Output: My name is Jesh", and "There are 4 words in the sentence inputted."

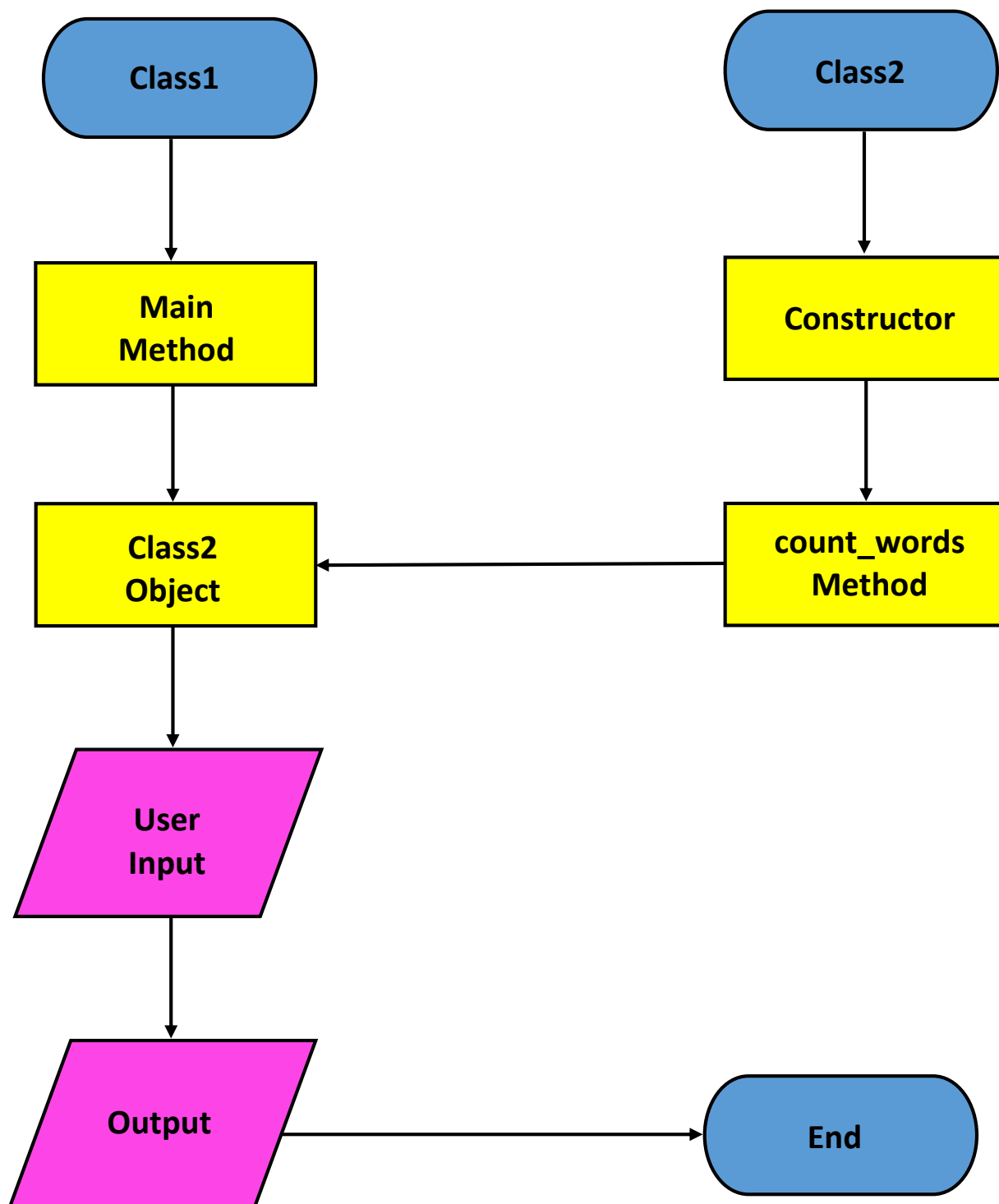
```
File Edit Selection View Go Run Terminal Help
QuestionOne.java - Visual Studio Code

QuestionOne.java 3 X
C: > Users > Jesh > Desktop > Java Q1 > QuestionOne.java > ...
1 import java.util.Scanner;
2 import java.util.StringTokenizer;
3
4 class Class1{
5     public static void main(String[] args){
6
7         Scanner inputKeyBoard = new Scanner(System.in);
8         System.out.print("Please enter your sentence: ");
9         String userInput = inputKeyBoard.nextLine();
10        System.out.println("Output: " + userInput);
11
12        Class2 object = new Class2();
13        int totalWords = object.count_words(userInput);
14
15        System.out.println("There are " + totalWords + " words in the sentence inputted.");
16    }
17}

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\Jesh> & 'C:\Program Files\Java\jdk-17.0.2\bin\java.exe' '-enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Jesh\AppData\Local\Temp\vscodesws_84a79\jdt_ws\jdt.ls-java-project\bin' 'Class1'
Please enter your sentence: My name is Jesh
Output: My name is Jesh
There are 4 words in the sentence inputted.
PS C:\Users\Jesh>
```

1.3) CONSTRUCT A FLOWCHART FOR CLASS1 AND CLASS2 BOTH COMBINED. (10)



## QUESTION TWO

**CREATE A JAVA PROGRAM THAT WILL DISPLAY THE FIRST 40 PENTAGONAL NUMBERS. (20)**

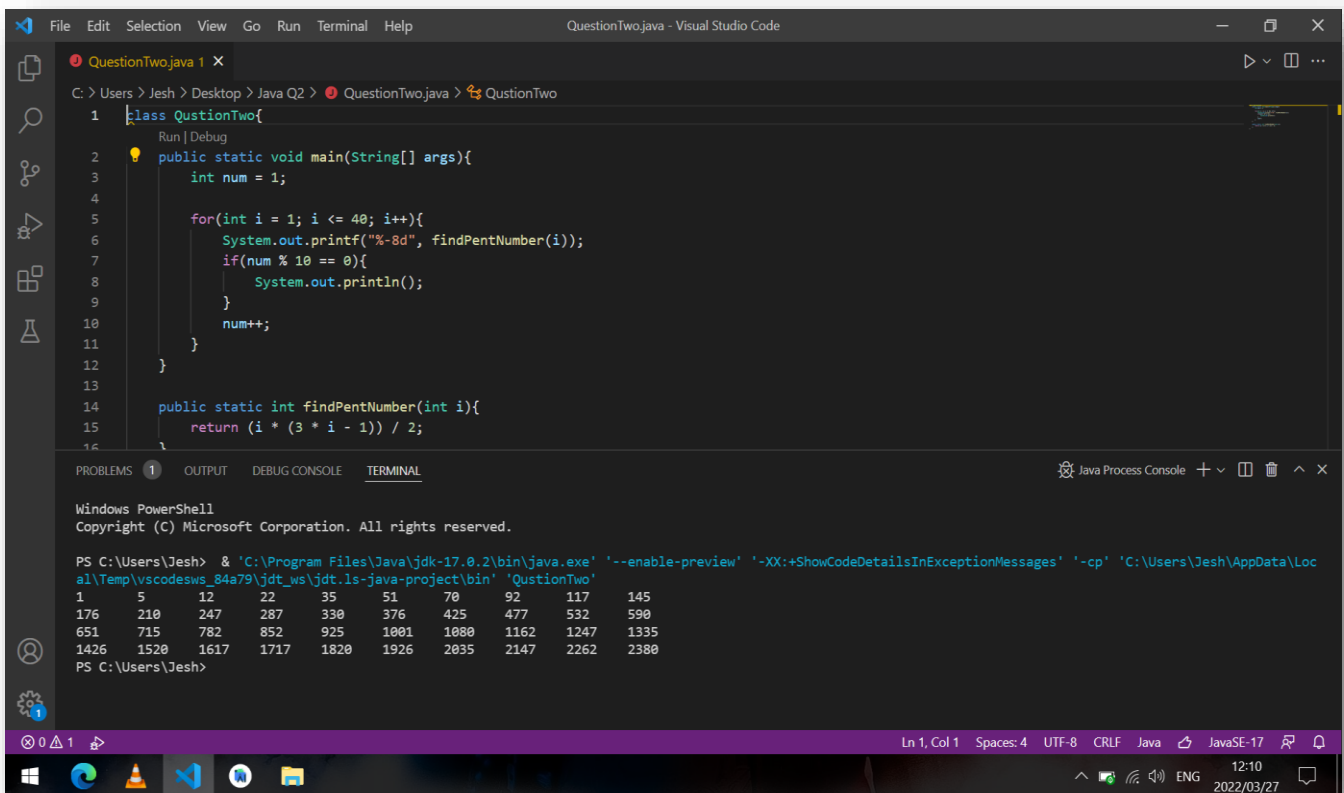
**Below is the entire code to execute the program:**

```
class QustionTwo{
    public static void main(String[] args){
        int num = 1;

        for(int i = 1; i <= 40; i++){
            System.out.printf("%-8d", findPentNumber(i));
            if(num % 10 == 0){
                System.out.println();
            }
            num++;
        }
    }

    public static int findPentNumber(int i){
        return (i * (3 * i - 1)) / 2;
    }
}
```

**Below is the output of the above code:**



The screenshot shows the Visual Studio Code editor with the Java code from the previous block. The terminal window at the bottom displays the output of the program, which is the first 40 pentagonal numbers arranged in four rows of ten. The numbers are: 1, 5, 12, 22, 35, 51, 70, 92, 117, 145; 176, 210, 247, 287, 330, 376, 425, 477, 532, 590; 651, 715, 782, 852, 925, 1001, 1080, 1162, 1247, 1335; 1426, 1520, 1617, 1717, 1820, 1926, 2035, 2147, 2262, 2380.

```
PS C:\Users\Jesh> & 'C:\Program Files\Java\jdk-17.0.2\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Jesh\AppData\Local\Temp\vscodesws_84a79\jdt_ws\jdt.ls-java-project\bin' 'QuestionTwo'
1      5      12      22      35      51      70      92      117      145
176    210    247    287    330    376    425    477    532    590
651    715    782    852    925    1001   1080   1162   1247   1335
1426   1520   1617   1717   1820   1926   2035   2147   2262   2380
PS C:\Users\Jesh>
```

## QUESTION THREE

**WRITE A JAVA PROGRAM THAT WILL COMPUTE THE FUTURE INVESTMENT VALUE AT A GIVEN INTEREST RATE FOR A SPECIFIED NUMBER OF YEARS. THE JAVA PROGRAM SHOULD HAVE A MINIMUM OF TWO CLASSES. (40)**

### **3.1) Requirements for the first class:**

- ❖ It should have the main method (starting point). (5)
- ❖ It should have the object of **classB**. (5)
- ❖ It should also call the void method named **futureInvestmentValue** (**double investmentAmount, double monthlyInterestRate, int years**). (15)

### **3.2) Requirements for the second class:**

- ❖ A get method named **count\_words**(String str). (15)

**Below is the entire code to execute the program:**

```
import java.util.Scanner;
import java.util.StringTokenizer;

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

        int number = 1;

        Scanner inputKeyBoard = new Scanner(System.in);

        System.out.print("Input the investment amount: ");
        double investAmount = inputKeyBoard.nextDouble();

        System.out.print("Input the rate of interest: ");
        double rate = inputKeyBoard.nextDouble();

        System.out.print("Input number of years: ");
        int years = inputKeyBoard.nextInt();

        rate *= 0.01;

        ClassB calc = new ClassB();

        System.out.println("Years:      Future Value:");

        for(int i = 1; i <= years; i++){
```

```

        System.out.printf(i + "          " + "%" + ".2f\n", calc.futureInvestmentValue((investAmount),
            rate/12, i));
    }

}

}

class ClassB{

    public static double futureInvestmentValue(double investmentAmount, double monthlyInterestRate,
        int years){
        return investmentAmount * Math.pow(1 + monthlyInterestRate, years * 12);
    }

}

```

**NB:** It was required to have a method called `count_words(String str)` in `classB`. However, this program requires the user to enter only integers and not words. This means that we cannot use the `count_words(String str)` method. Therefore I have moved the method called `futureInvestmentValue` into `classB` in order to show the same principle of calling a function from an external class into the main function.

*Below is the output of the code after gathering data from the user:*

```

C:\> Users > Jesh > Desktop > Java Q3 > QuestionThree.java > ClassA > main(String[])
12 double investAmount = inputKeyBoard.nextDouble();
13
14 System.out.print("Input the rate of interest: ");
15 double rate = inputKeyBoard.nextDouble();
16
17 System.out.print("Input number of years: ");
18 int years = inputKeyBoard.nextInt();
19
20 rate *= 0.01;
21
22 ClassB calc = new ClassB();
23
24 System.out.println("Years:          Future Value:");
25
26 for(int i = 1; i <= years; i++){
27     System.out.printf(i + "          " + "%" + ".2f\n", calc.futureInvestmentValue((investAmount), rate/12, i));
28 }

```

```

PS C:\Users\Jesh> & 'C:\Program Files\Java\jdk-17.0.2\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Jesh\AppData\Local\Temp\vscodesws_84a79\jdt_ws\jdt.ls-java-project\bin' 'ClassA'
Input the investment amount: 28000
Input the rate of interest: 7
Input number of years: 6
Years:          Future Value:
1              30024,12
2              32194,57
3              34521,92
4              37017,51
5              39693,51
6              42562,95
PS C:\Users\Jesh>

```

## REFERENCES

- Admin. (2020, December 11). *Java Constructors*. Retrieved from W3schools:  
[https://www.w3schools.com/java/java\\_constructors.asp](https://www.w3schools.com/java/java_constructors.asp)
- Admin. (2020, August 23). *Java User Input (Scanner)*. Retrieved from W3schools:  
[https://www.w3schools.com/java/java\\_user\\_input.asp](https://www.w3schools.com/java/java_user_input.asp)
- Admin. (2021, June 24). *Java For Loop*. Retrieved from W3schools:  
[https://www.w3schools.com/java/java\\_for\\_loop.asp](https://www.w3schools.com/java/java_for_loop.asp)
- Admin. (2021, November 28). *Java Methods*. Retrieved from W3schools:  
[https://www.w3schools.com/java/java\\_methods.asp](https://www.w3schools.com/java/java_methods.asp)
- Admin. (2022, February 12). *Java Classes and Objects*. Retrieved from W3schools:  
[https://www.w3schools.com/java/java\\_classes.asp](https://www.w3schools.com/java/java_classes.asp)
- Admin. (2022, February 18). *Java Math*. Retrieved from W3schools:  
[https://www.w3schools.com/java/java\\_math.asp](https://www.w3schools.com/java/java_math.asp)
- Admin. (2022, January 13). *Java Variables*. Retrieved from W3schools:  
[https://www.w3schools.com/java/java\\_variables.asp](https://www.w3schools.com/java/java_variables.asp)