

Assessment item 1—Java Console Program

Due date: Week 7 T317 – Midnight, Friday 5 January 2018 **ASSESSMENT**

Refer below for complete assessment item 1 requirements
(Assignment One)

Weighting: 15%

Length: N/A

1

Objectives

This assessment item relates to the course learning outcomes as in the Unit Profile.

Details

For this assignment, you are required to develop **Java Console Programs** to demonstrate you can use Java constructs including input/output via a command line and using GUI dialogs, Java primitive and built-in data types, Java defined objects, selection and looping statements, methods, and various other Java commands. Your program must produce the correct results.

You are only allowed to use techniques which have been covered in the first five weeks of the subject and within the assignment literature, you must use a **Scanner** object for console input and no advanced data structures like arrays will be used.

What to submit for this assignment

The Java source code:

You will be able to complete the assignment in weekly parts in which you will produce five java source files. (More details below)

Week1.java, Week2.java, Week3.java, Week4.java and Week5.java.

Once you have completed all of the programs and you are ready to submit, compress all source files into a single zip file for submission, **do not include your report in the zip file**. Only submit a zip not a rar file

- **Ass1.zip**

Also submit a report including, how long it took to create the programs (approximately), any problems encountered and screen shots of the output produced. (Use Alt-PrtScrn to capture just the console window or your dialogs and you can paste it into your Word document) You should test every possibility in the program and annotate your test screen shots.

- **ReportAss1.docx**

You will submit your files by the due date using the “**Assignment 1**” link on the Moodle unit website in the **Assessment Block** or in the relevant week.

Assignment specification

This assignment will require you to write small five programs, do not panic! They will be small programs which will cover the first five weekly topics. Usually students were required to write one largish program to demonstrate the topics for the first five weeks. Students get themselves into trouble when the first assignment is due as they have not practiced the basics skills necessary to complete the assignment. With the assignment divided into five programs you can complete each exercise as we cover the weekly topics, do not let yourself fall behind.

General Instructions

Each program must contain a header comment which contains: Your name and student number, the name of the file, the date and a brief description of the purpose of the program:

```
// Programmer: Eric Gen S01234567
// File: Week1.java
// Date: January 5 2018
// Purpose: COIT11222 assignment one question one T317
// Use println method to print initials using asterisks
```

All programs will be aligned and indented correctly, and contains relevant comments for declarations and statements. All variables and objects will be declared with a meaningful name and use lowercase camel notation:

```
String personName;
```

All code will be contained within a main method except for question five when a method will be created and used.

For this assignment you will not worry about checking numeric ranges or data types.

Refer to a Java reference textbook and the unit and lecture material (available on the unit WEB site) for further information about the Java programming topics required to complete this assignment.

Check the marking guide (last page) to ensure you have completed every task. You need to match all outputs exactly as the sample screenshots shown below.

Distance and Melbourne students can email questions directly to me, other metro campus students should seek help from your local tutor, you can still contact me if it is urgent, I usually respond to emails very promptly.

Good luck --- Bruce McKenzie COIT11222 unit coordinator term 3 2017

b.mckenzie@cqu.edu.au

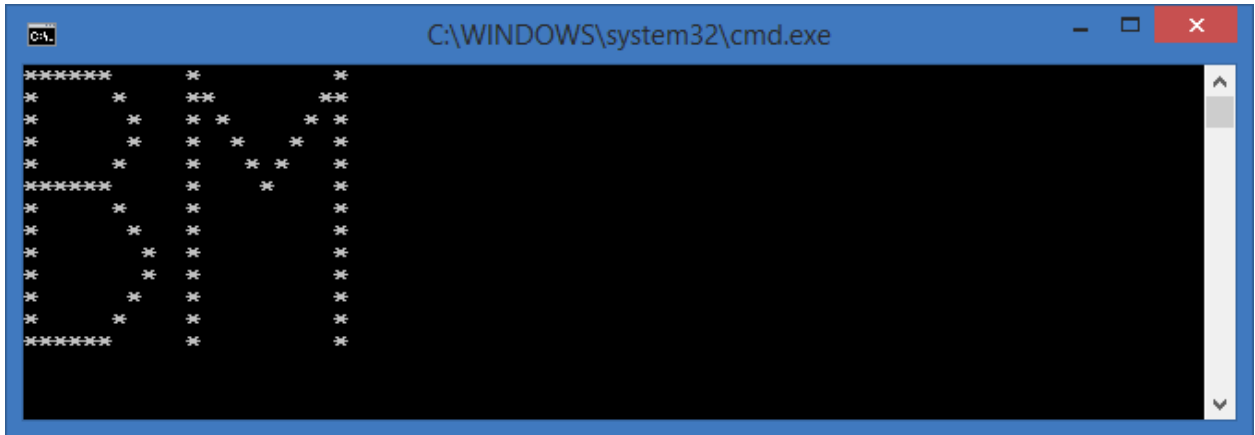
Question one (week one topic). Writing output to the screen.

Once you have written your first “Hello World” program you will be able to complete question one.

Implementation

Create a class called **Week1** (file: Week1.java) and within it a main method.

Use the command `System.out.println(" ");` to print out the first initial of your first and last names as a matrix of asterisks. For example this is my first and last initials printed.



The first line of asterisks is printed with this command:

```
System.out.println("***** * *");
```

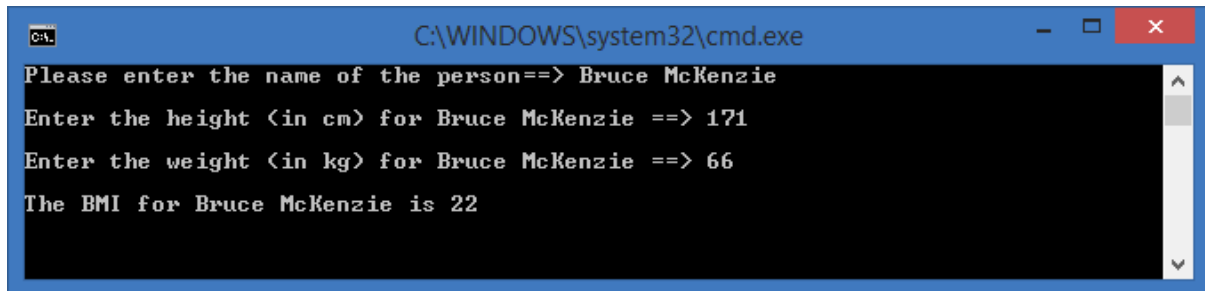
You may need to use some graph paper to plot where you need to print your asterisks.

If you like you could submit a picture. An attempt at Mickey Mouse! Just do your initials as it takes a while to create a picture.



Question two (week 2 topics) Input of data types and arithmetic expressions

This program will prompt for and read in a person's name using a Scanner object and output the name in another prompt for the height (in centimetres) for the person then the program will prompt for the weight of the person in kilograms. The program will read the height and weight for the person and output the Body Mass Index (BMI) for the person (the formula is below). You need to replicate the output as shown below.



```
C:\WINDOWS\system32\cmd.exe
Please enter the name of the person==> Bruce McKenzie
Enter the height <in cm> for Bruce McKenzie ==> 171
Enter the weight <in kg> for Bruce McKenzie ==> 66
The BMI for Bruce McKenzie is 22
```

Implementation

Create a class called **Week2** (file:Week2.java) and within it a main method as per question one.

Import the Scanner class i.e.

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

Within your main create two Scanner objects named **inText** and **inNumber**. One for reading text and the other for reading the numbers, it does not really matter here to have separate Scanner objects but there will be problems later when reading a series of text and numbers (see text pg 81).

Create a prompt using `System.out.print()`; To ask the user for the name of the person.

Declare a String object **personName** to store the person's name and use your **inText** Scanner object and the inbuilt method `inText.nextLine()`;

The person name is now stored in the String object **personName**.

We can now create a prompt using the person's name to ask for their height in centimetres.

Hint: you can join variables and strings using the concatenation operator +

```
"Enter the height(in cm) for " + personName + "etc... "
```

Declare a double variable to store the height and use your **inNumber** Scanner object and the inbuilt method `inNumber.nextDouble()`; to read the number.

Repeat this to read in the person's weight in kilograms.

Declare an integer variable to represent the BMI for the person.

Create an arithmetic expression to calculate the BMI as follows.

How to calculate the BMI

The formula for the BMI is:

$BMI = \text{weight in kilograms} / (\text{height in metres} * \text{height in metres})$

You will need to convert the centimetres into metres for this calculation.

```
height = height / 100;
```

You will get an error if you try and assign the double expression to the integer BMI variable so you must use the (int) cast operator in the assignment statement. Ensure you cast the whole expression.

Hint: place the whole expression in parenthesis and put the cast operator on the outside.

Print out the result as above using the concatenation operator.

Question three (week three topics) Decision statements

Use the pseudo code below to create a class **Week3** (file: Week3.java) and a main method which uses decision statements.

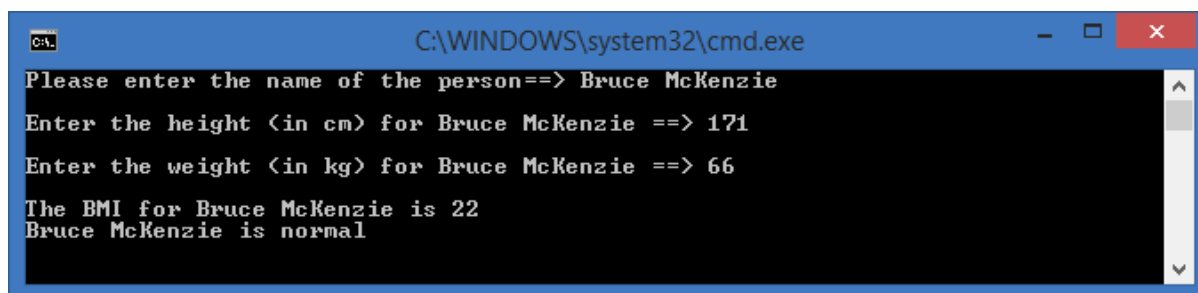
```
WRITE "Please enter the name of the person ==> "  
READ name  
WRITE "Enter the height (in cm) of " + name + " ==> "  
READ height  
WRITE "Enter the weight (in kg) of " + name + " ==> "  
READ weight
```

```
Calculate the BMI of the person // from week 2
```

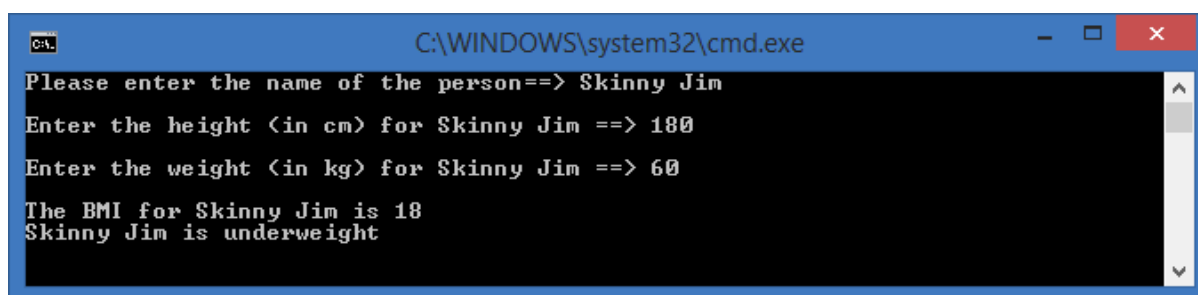
```
IF BMI is less than 19 THEN  
    rating is assigned "underweight"  
ELSE IF BMI is less than 25 THEN  
    rating is assigned "normal"  
ELSE IF mark is less than 30 THEN  
    rating is assigned "overweight"  
ELSE  
    rating is assigned "obese"  
ENDIF
```

```
WRITE "The BMI for " + name + " is " + BMI  
WRITE name + " is " + rating
```

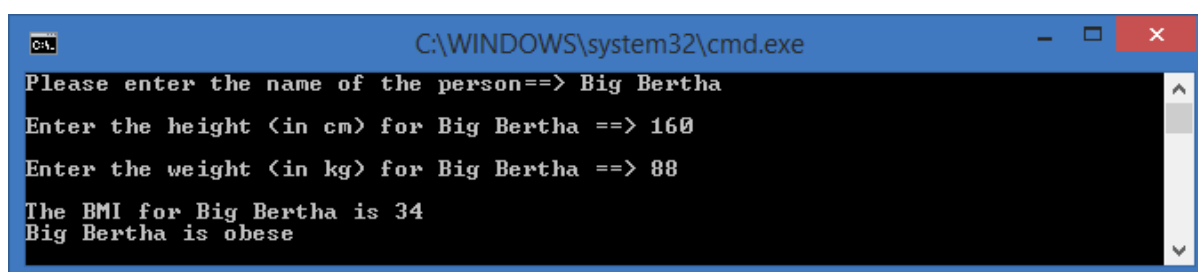
This program is demonstrating the use of “if” statements in decision making. The program will read one person’s name, height and weight and calculate their BMI, and use the if statements to assign the rating to a String object. After the if statements the program will print the person’s name, BMI and rating grade (see sample output below). **Note: you must use constants for the numbers in the if statements.**



```
C:\WINDOWS\system32\cmd.exe
Please enter the name of the person==> Bruce McKenzie
Enter the height <in cm> for Bruce McKenzie ==> 171
Enter the weight <in kg> for Bruce McKenzie ==> 66
The BMI for Bruce McKenzie is 22
Bruce McKenzie is normal
```



```
C:\WINDOWS\system32\cmd.exe
Please enter the name of the person==> Skinny Jim
Enter the height <in cm> for Skinny Jim ==> 180
Enter the weight <in kg> for Skinny Jim ==> 60
The BMI for Skinny Jim is 18
Skinny Jim is underweight
```



```
C:\WINDOWS\system32\cmd.exe
Please enter the name of the person==> Big Bertha
Enter the height <in cm> for Big Bertha ==> 160
Enter the weight <in kg> for Big Bertha ==> 88
The BMI for Big Bertha is 34
Big Bertha is obese
```

Implementation

Create a class Week3 and a main method and also create Scanner objects as per question 2.

Read in the person’s name, height and weight and calculate their BMI.

Declare a String object **rating** to store the rating string within the if statements.

Use nested if...else if statements to assign the correct rating string.

Finally output the name, BMI and rating as per the examples above.

Follow the above pseudo code, enter a line or two of code and compile, always ensure you are working with clean (error free) code.

Question four (week four topics) Repetition while and for loops

Create a class **Week4** (file:Week4.java) to demonstrate the use of a repetition statement.

Implementation

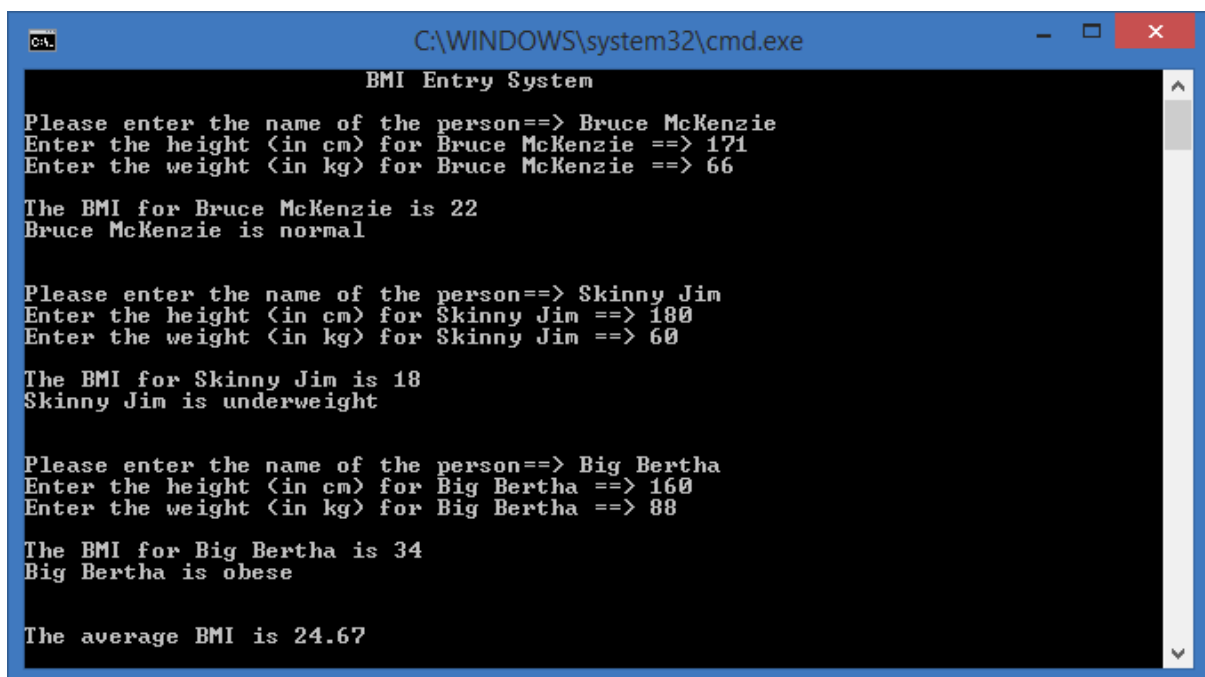
Using your solution to question three and a `while` or `for` loop, repeat the entry of person names, heights and weights **N** times where **N is the largest digit in your student ID**, if your largest digit is less than three then let $N = 3$. Hint: use $N = 3$ while testing and submit using the correct N value.

N will be declared as an integer constant using the `final` keyword.

You are to print a title before the input of the persons' names, heights and weights (see sample output).

Ensure you are using a separate Scanner objects for reading numbers and text. (Why?)

When all of the persons' names, heights and weights have been entered the average of the BMIs will be reported. Please note you do not need to store the data in an advanced structure such as an array. You will need to have an integer variable to add up the BMIs to calculate the average.



```
C:\WINDOWS\system32\cmd.exe
BMI Entry System
Please enter the name of the person==> Bruce McKenzie
Enter the height <in cm> for Bruce McKenzie ==> 171
Enter the weight <in kg> for Bruce McKenzie ==> 66
The BMI for Bruce McKenzie is 22
Bruce McKenzie is normal

Please enter the name of the person==> Skinny Jim
Enter the height <in cm> for Skinny Jim ==> 180
Enter the weight <in kg> for Skinny Jim ==> 60
The BMI for Skinny Jim is 18
Skinny Jim is underweight

Please enter the name of the person==> Big Bertha
Enter the height <in cm> for Big Bertha ==> 160
Enter the weight <in kg> for Big Bertha ==> 88
The BMI for Big Bertha is 34
Big Bertha is obese

The average BMI is 24.67
```

Your average BMI calculation has to produce a floating point result. (To get a floating point result you will need to promote one of the operands to a double i.e. `average = total * 1.0 / N`) To format your average to two decimal places you can use the `printf` statement with a format string.

```
System.out.printf("%.2f", average);
```

You could also use `String.format`:

```
System.out.println(String.format("%.2f", average));
```

Question five (week five topics) Methods and GUI I/O

Create a class **Week5** (file:Week5.java) by using your solution to question four. This question is identical to question four as the program will read in N person names, height and weights and calculate the BMI and rating, however we are going to create a method and we will be using GUI dialog boxes for our I/O.

Implementation

Methods

You will create a value returning method which will accept the height and weight as a parameter.

Use the following method header:

```
private static String getRating(int bmi)
```

Copy and paste your “if else” code for calculating the rating from the BMI into the body of our new method `getRating`. Use the `return` statement to return the grade string.

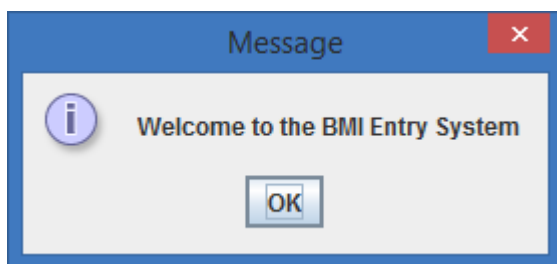
You can now use your method in the main method loop.

```
String rating = getRating(bmi);
```

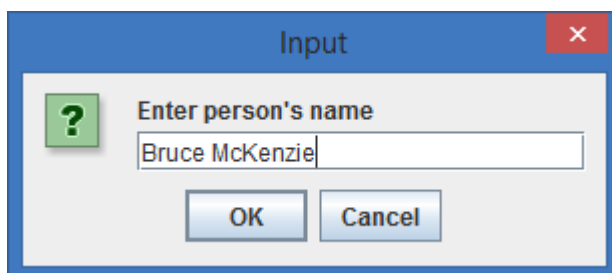
GUI I/O

We will revisit the week two topic using `JOptionPane` for accepting GUI input and outputting information.

First we will output a welcome message using `JOptionPane.showMessageDialog`. (Replace your console print output)

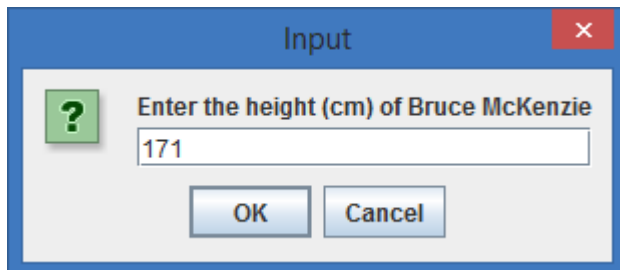


Next we will replace the Scanner objects by using `JOptionPane.showInputDialog`.



The showInputDialog method will return the string entered into the dialog text field

```
String str = JOptionPane.showInputDialog(null, "Prompt");
```

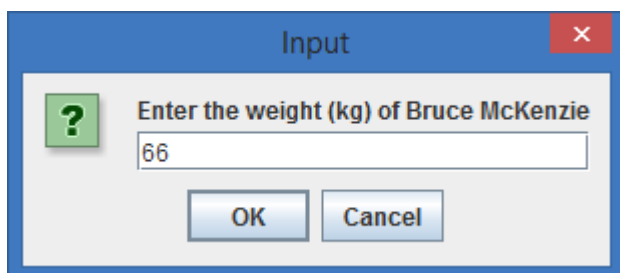


We receive input from the dialog as a string, in order to convert strings to doubles we need the Double wrapper class and parseDouble method (text pg 370).

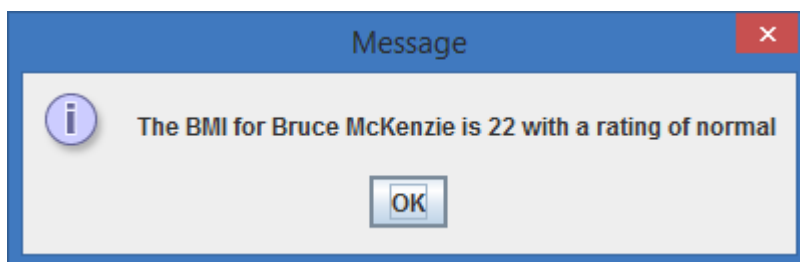
```
double aDouble =
```

```
Double.parseDouble(JOptionPane.showInputDialog(null, "Prompt"));
```

Do the same for entering the weight.

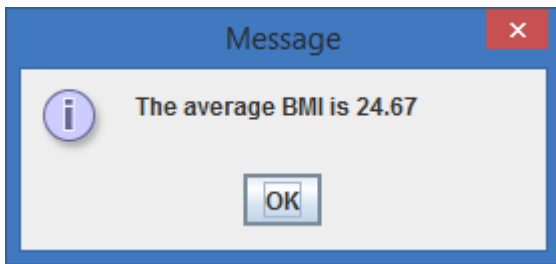


After reading in and converting the height and weight input to a double you can use these variables to calculate the person's BMI, and then use the BMI value to call your user defined method `getRating(BMI)`. Using the returned rating string you can output this information using: `JOptionPane.showMessageDialog`



You can use the String concatenation operator `+` to join the String objects with your text in the message dialog.

When N students have been entered as per question 4 you will report the average BMI.



You can use `String.format("The ... %.2f", average)` to format your average to two decimal places.

The marking scheme is on the next page.

Marking Scheme

	Total number of marks – 15	
1	Code in general	
	Code is indented and aligned correctly, layout including vertical white space is good	0.5
	Code has header comment which includes student name, student ID, date, file name and purpose of the class	0.5
	Code is fully commented including all variables	0.5
	Variables have meaningful names and use camel notation	0.5
	Variables are the correct type	0.5
2	Question one	
	Output as per specification	1
3	Question two	
	String read correctly using Scanner object	0.5
	The doubles are read correctly using a Scanner object	0.5
	The BMI is computed and displayed correctly	0.5
	Output is formatted correctly (matches sample output)	0.25
4	Question three	
	If else statement correct and constants are used	1
	Correct rating is produced	0.25
	Output is formatted correctly (matches sample output)	0.25
5	Question four	
	Constant N used equal to highest digit in student ID	0.5
	N person names, heights and weights are read in a loop	1
	Program title "BMI Entry System" printed	0.25
	Rating printed for all persons	0.25
	Average is calculated and printed correctly to two decimal places	1
	Output is formatted correctly (matches sample output)	0.25
6	Question five	
	Method implementation	0.5
	String value returned from method correctly	0.25
	Method call correct	0.5
	GUI welcome message	0.25
	String read correctly from GUI Input dialog	0.25
	Height and weight read correctly from GUI Input dialog and converted to doubles	0.5
	N person names, heights and weights are read in a loop	0.25
	Average is calculated and printed correctly to two decimal places	0.25
	Dialogs appear as per specification (matches sample output)	0.25
7	General	
	Correct files submitted including types and names (zip and Word)	0.5
	Only techniques covered during weeks 1-5 are used	0.5
8	Report	
	Report presentation and comments including how long it took and any problems encountered	0.5
	Screen shot(s) of testing	0.5