Guided Exploration 02: Classes, Objects, Methods, Decisions, Algorithms

**Points:** 30 (see rubric in canvas)

**Due Date:**  See Canvas.

Work will be accepted up to 24 hours after the due date with a 10% penalty. Meaning if you turn it in at 12:01 am of the next day you will be deducted 10% of the total points from your score. **If the assignment is more than 24 hours late, it will be a 0.**

Purpose: Guide your learning to meet the following objectives for ABET 1 and 5

* Analyze and understand computing problems by breaking down complex problems into manageable parts and understand its components.
* Identify and apply relevant theories, principles, and methods from computing such as object orientation, algorithms, coding structures, software engineering practices, and technology tools.
* Demonstrate the ability to collaborate effectively contributing to discussions in class and in teams

**Effort: Collaborative and Individual**

* You can collaborate but individually **write your own code and answer the questions in your own words** . When you ask questions and explain to others you get a deeper understanding.
* **Remember: Only use concepts we have discussed in lectures. You can use the System, Scanner, Math and Character classes. Do not use AI to answer the questions or to create code.** Handing in something that is not your own work is not beneficial to your learning and will be a violation of ([Academic Integrity](https://docs.google.com/document/d/1KsBBSZ3b227g1t8ciF3lwJBGnpKNCV9nEXRFmTLTZdY/edit#heading=h.1k4ycmhenia3)) .
* You can use it to clarify information or get help with syntax errors.

**Deliverables:** Upload each file separately and not as a zip file

* Upload this document as a pdf or word document with your answers
* Upload your LastnameGE02StringsChars.java file (not your class file)
* Upload your technical documentation as a pdf or word document with your answers

[Part 1: Explain and Analyze](#_4kqqreg5puc)

[1.1 Classes, Objects, Methods and Decision Structures](#_lnqz0m7jv069)

[Part 2: Apply](#_4q84c1d4y6lu)

[2.1 M02 L05 BMI Lab Program](#_601obpmk994h)

[2.2 Create Program](#_sf0usmgpmqyb)

[Part 3: Technical documentation](#_ycma38s0k73f)

[4 Reflection](#_sokkf1ov5gyk)

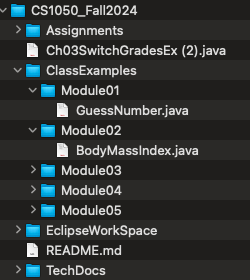
# Part 1: Explain and Analyze

Go back to the lectures and the code I had you explore in class to complete the following.

Put this document in a folder in your github repository.

## 1.1 Classes, Objects, Methods and Decision Structures

Import into Eclipse the guessnumber.java in module 1 if you haven't yet. This is in the ClassExamples you downloaded when you set up your environment.



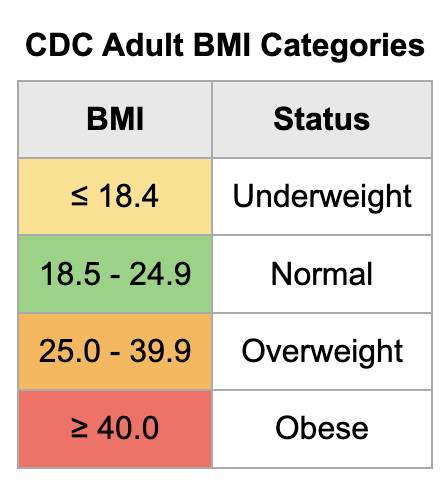
* As you answer the following add comments to your code and you can copy and paste small code snippets to answer the questions.
* You will also need to go to the lectures to find other code examples that are not in this example. Remember to look at the code links in the lectures that you can run the animations and compile/run
* Remember to commit and push

| 1.1.1 Describe packages, classes and import. |
| --- |
| 1.1.2 Explain how to read an integer input from the keyboard by using class to create object instances and calling methods. Fix the warning by calling the close() method at the end of the program. |
| 1.1.3 Use your IDE to show how to use the API to see the description of the built-in Math method to raise to a power: Math.pow(). Put a screenshot. |
| 1.1.4 Explain the Math.random method. Update the code to use a formula to find an integer between a MIN and MAX number as discussed in the lecture. |
| 1.1.5 Explain the difference in using the Math class and the Scanner class. |
| 1.1.6 Explain how information is printed to the console in the GuessNumber.java code. |
| 1.1.7 Explain boolean values  Explain the use of  >, <, >=, <=, ==, !=  Explain the difference between == and = |
| 1.1.8 Explain the parts of a two way selection structure if/else. Include an examples from other code. |
| 1.1.9 Explain the decision structure that is used in the GuessNumber.java code. |
| 1.1.10 Explain scope in terms of where you declare variables in if else blocks or outside of if else block in the main method. Include code examples from other code. |
| 1.1.11 Explain the difference of when you would use a nested if or a multiway if. Include code examples |
| 1.1.12 Explain the difference between logical operators &&, || and !  Include examples from code. |
| 1.1.13 Explain using a switch and the parts of the code with examples. What situations you can use a switch instead of an if else and when you can’t. |
| 1.1.1.4 Commit and push and show your versions of the GuessNumber.java you have in your github remote repository. |
| 1.1.6 Go to section [Part 3: Technical documentation](#_ycma38s0k73f) and add information to your technical document using information from above |

# Part 2: Apply

## 2.1 M02 L05 BMI Lab Program

* Read M02 L05 slide 24. Complete the lab.
* Update your code so that after the BMI is displayed the category for that BMI is displayed. Use the following categories. Think about what would be the best decision structure algorithm to use for this problem. Nested if, multi-way if or switch statement.



Example output

| This program will calculate your body mass index (BMI)  Enter your weight in pounds: 150.5  Enter your height in inches: 66  Weight: 150.5 Height: 66.00  Your body mass index (BMI) is 24.3  Your category: normal |
| --- |

| 2.1.1 What algorithm did you choose to display the correct category status and why? Why can’t you use a switch statement? |
| --- |
| 2.1.2 Copy and paste the code you created to display the correct category. You do not need to copy the entire code. |
| 2.1.3 Create test data to make sure you test each category condition. I filled in one below.  Put a link to a website that calculates the BMI to come up with your test data.  Link:   | Test Data | Output | | --- | --- | |  | Underweight | | 150.5 lbs  66 inches  Bmi 24.3 | Normal | |  | Overweight | |  | Obese |   2.1.4 Then run the debugger to show your test data will step into the correct condition. Put a screenshot showing the variables while using the debugger. |
| 2.1.5 Add comments to your code so you can use this in your technical document. Commit and push. Put a screenshot showing your code in your github remote repository. You do not need to upload this java file. |

## 2.2 Create Program

Create a java class in your project called LastnameGE02StringsChars. Do not use AI to write your code. Come see me if you are having problems writing the code.

**Problem Statement**

Write a program and add comments that does the following:

1. Prompts the user and reads in the first name and last name of two students as Strings.

| Enter Student 1 first name: Stack  Enter Student 1 last name: Terror  Enter Student 2 first name: Memory  Enter Student 2 last name: Leak |
| --- |

1. Compares the last names of both students using string compareToIgnoreCase and an if/else statement to determine which name should be printed first. For example:

| Students in alphabetical order by last name:  Memory Leak  Stack Terror |
| --- |

1. Extract the first character of both the first and last names and store in separate char variables for each students first initial and last initial
2. Creates the initials of each student by concatenating the first character of their first and last names with periods. Store this in a String.
3. Outputs the initials of both students. For example:

| Initials:  Student 1: S. T.  Student 2: M. L. |
| --- |

| 2.2.1 Explain the difference between declaring a String and a char data type. Include examples. |
| --- |
| 2.2.1 Explain how to read in a string and how to read in a char. Include examples. |
| 2.2.3 Explain what charAt(0) is doing. Include examples. |
| 2.2.4 Explain the string method to compare strings. Include examples. |
| 2.2.5 Explain what concatenate means and how you do it. Include examples. |
| 2.2.6 Explore two other string functions. Explain and include examples. |
| Upload your java source file when you submit and make sure you put it in your github remote repository. Update your technical documentation. |

# Part 3: Technical documentation

| Go to your technical document and create headings for module 2. Here are some examples of headings.   * Look at what you completed for Part 1 and 2 above. That might be good to include. * What other information would be valuable from lectures   Heading 1: “Module 2: Predefined Classes, Methods and Decisions Structures”  Heading 2: “Classes, Objects and Methods”   * Importing classes * Creating classes and accessing methods   + What is a class   + What is a method - method signature, parameters (formal and actual), return values   + Classes we use where you don’t create an object examples where you do create an object.     - Math class and methods such as generate a random number and the algorithm to generate an integer between two integer values   + Classes we use where you do create an object to access methods     - Scanner class and Methods for Reading in from the keyboard int, double, char, String   + String classes and methods   Heading 2: “Conditions”   * Include different types of conditions used to decide what code should be run * What algorithms for conditions are used for different situations |
| --- |

## 4 Reflection

| 4.1 Identify one specific challenge you faced in this exploration. How did you address it using the strategies discussed (e.g., desirable difficulties, learning vs. performing)? |
| --- |
| 4.2 Explain how you ensured the AI usage was used to help your learning and did not hinder your learning for this exploration. |
| 4.3 What was the most challenging part of this exploration, and why? |
| 4.4 How did you manage your time to complete each section after lectures? |

Read what the deliverables are for this assignment and submit them all to canvas. You can submit at different times but pay attention to the due date and the late date.