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)

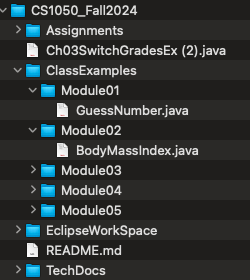
# Part 1: Explain and Analyze

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.



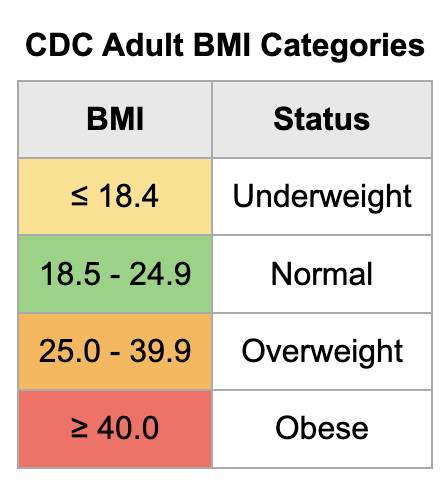
* As you answer the following add comments to your code and you can copy and paste small code snippets to answer the questions.
* You will also need to go to the lectures to find other code examples that are not in this example. Remember to look at the code links in the lectures that you can run the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

|  |
| --- |
| 1.1.1 Describe packages, classes and import.    Packages – A package is a sort of organization of classes. For example, java.util is a package and contains the .scanner class.    Classes – Classes are used in unison with .java projects. These are essentially blueprints or guides for the code they enclose data, and methods.    Methods – Methods are instructions for the code, these are blocks that run once called upon. For example, the “main” argument is a method for the whole block, .charAt(0) is a method using the string that is for that one line or block of code. |
| 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.      Using the class, the method “main” can be called upon. This will be where your code is encapsulated and run. The class “scanner” is a default class, however, the java.util package must be imported. Using the scanner class, a new object can be created, this object will be a system scanner that is used to register the input from your keyboard. |
| 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.    The default method, math, can be used to create an exponential number. For example, here Math.pow(10,2) would be 10 squared equating to 100. |
| 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.  The Math.random method is a way to generate a random value for a variable.    [Java Doc Operators](https://docs.oracle.com/javase/tutorial/java/nutsandbolts/operators.html)  To create a formula for MIN and MAX values you must use logical operators and relational operators. Logical operators being “&&” which means “and”, and relational operators being “< or >” being “greater than” and “less than”. These operators will help establish a minimum value, so, if guess < 0, it is an error. The same goes for maximum number, if guess > 100, it is an error. |
| 1.1.5 Explain the difference in using the Math class and the Scanner class.  The math class is used for mathematical equations, it is a default class that does not need to be imported. The scanner class is used to register inputs, the scanner class is a default class, however, it must be imported through the java.util package. |
| 1.1.6 Explain how information is printed to the console in the GuessNumber.java code.  The system.out.println tells the code to print text if the “if statement”, “else statements”, or “else statements” are true. So, if the guess is above the maximum number, 100, it will print “Your guess is out of the boundary 0-100”, on the other hand, if the correct number is guessed, the code will print “Yes, the number is” and then the correct guess. |
| 1.1.7 Explain boolean values  Boolean values are true or false values, they are a primitive data type with only one possible value attached to it at a time, however, there are two possible values, true or false.  Explain the use of  >, <, >=, <=, ==, !=  All these operators are comparison operators, they are used to comparing two values, if a value is greater than the other, equal to the other, etc.  Explain the difference between == and =  “==” indicates equal to as a comparison operator, “=” is used to assign values to a variable or constant. |
| 1.1.8 Explain the parts of a two way selection structure if/else. Include an examples from other code.    If/else statements are used when only one thing about a variable can be true at a time. So, for example, if the finalg value is not >= 90 or <= 100, it would be a B, if it is not a B, it could be a C, if not a C or a D, it is an F. If a condition is false, the next condition is tested. |
| 1.1.9 Explain the decision structure that is used in the GuessNumber.java code.  [Lawrence on Decision Structures](https://www2.lawrence.edu/fast/GREGGJ/CMSC150/012Decisions/Decisions.html)  The decision structure is used to assign letter grade values to a percentage grade value. If and else if statements are used to help the program make decisions, in this case on what letter grade to output. It allows the program to run through each variable and the conditions it must be under. |
| 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.  When declaring variables within if else blocks, the variable’s scope is limited to that block. If the variable is declared outside of that block, in the main method, it can be used in the scope of the entire block that the method is referenced in. |
| 1.1.11 Explain the difference of when you would use a nested if or a multiway if. Include code examples  [Replit's Guess Animal](https://replit.com/@BerylHoffman/GuessAnimal#Main.java)  A nested if statement is an if statement inside another if statement, this means that the if statement inside the initial if statement will only be executed if the initial is true.  If the answer is yes to the question “is it a mammal?”, the question “is it a pet?” will run. If the initial question is no, the following question will not be asked at all. Then, after the initial nested if statements run, the else statements will run. |
| 1.1.12 Explain the difference between logical operators &&, and !  Include examples from code.    && - This refers to an “and” statement, which means two things need to be true for a statement  ! - This is a logical operator that means “not”. For example, while (guess != number)  != - refers to not equal, saying while the guess is not equal to the actual number.  || = This is a logical operator for “or”. This means one could be true, or the other could be true. One or both statements must be true, or else it’s false.  && = This is a logical operator that means “and”. For example, in the else if statement it contains >= 90 && (and) <= 100. This means the else if statement can only be true if the grade is above 90 and less than 100. |
| 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.  A switch allows the program to switch branches based on the value of a variable or expression. The switch stops running after a break statement is made, like input close.    You can use multiple values for a switch, and a good example of where it would be useful is what day of the week it is based on the number of days it has been, like a calendar. |
| 1.1.1.4 Commit and push and show your versions of the GuessNumber.java you have in your GitHub remote repository.    I made all the changes in one commit and push. |
| 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?  I chose to use <= >= operators in unison with if, and else if statements. This establishes a minimum and maximum value for every weight stage. This made the most sense in my head  ` |
| 2.1.2 Copy and paste the code you created to display the correct category. You do not need to copy the entire code.  if (bmi <= 18.4)  System.***out***.println(", You are underweight.");  else if (bmi >= 18.5 && bmi <= 24.9)  System.***out***.println(", You are a normal weight.");  else if (bmi >= 25.0 && bmi <= 39.9)  System.***out***.println(", You are overweight.");  else  System.***out***.println(", You are obese.");  This is a multiway else if example. The program will run through each variable condition by rendering one false, it goes on to the next until the value of BMI is in accordance with one of the weight stage. |
| 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 | | 140 lbs  76 inches  BMI 17.04 | Underweight | | 150.5 lbs  66 inches  Bmi 24.3 | Normal | | 160 lbs  57 inches  34.6 BMI | Overweight | | 200 lbs  50.2 inches  BMI 55.8 | 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.    String – A string is an object that can consist of multiple characters.  Char – A primitive data type that can only hold one character. |
| 2.2.1 Explain how to read in a string and how to read in a char. Include examples.  When reading in a string, it can be multiple characters, and it is defined by double quotations, “Example”.  When reading in a char, it can only be one character and must be defined by single quotations, ‘E’. |
| 2.2.3 Explain what charAt(0) is doing. Include examples.  charAt(0) is printing the character that is situated at the first position of text.  In this case, charAt(0) is printing the first character of both the first and last name. |
| 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.  Concatenate – Concatenate is connecting variables and quoted text in the print function. This is done using “+”. |
| 2.2.6 Explore two other string functions. Explain and include examples.    When adding a scanner, the same method of inputting a new int or double can be applied to string. String is basically an integer, so the user can input the value for a string.  String.compareToIgnoreCase allows the system to compare two values regardless of case, that is regardless of upper, or lowercase. |
| 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)?  I started out good in my time management. However, somewhere along the way I dropped the ball, I had to remind myself not to rush. |
| 4.2 Explain how you ensured the AI usage was used to help your learning and did not hinder your learning for this exploration.  I didn’t really use AI. I mostly used things like Javadoc and stack overflow. However, when I used these things, I used them to help me further develop my understanding code, not copy code and learn nothing. I wrote my code myself. |
| 4.3 What was the most challenging part of this exploration, and why?  The most challenging part of this exploration was probably the technical doc. Writing the code and making some comments and such is easy, so is filling out this guided exploration. The technical doc needs an updated table of contents, organization, proper structure, etc. |
| 4.4 How did you manage your time to complete each section after lectures?  In the beginning, I reviewed lecture before, and after the actual class. I got some work done on the GE early and later. Until the second half of the week, I dropped the ball. |

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.