Assessment Details and Submission Guidelines	
Trimester	T2 2022
Unit Code	BN111
Unit Title	Programming Fundamentals
Assessment Type	Assignment 2- Individual
Assessment Title	Interactive Programs
Purpose of the assessment (with ULO Mapping)	This assignment assesses the basic concepts of programming; students should be able to demonstrate their achievements in the following unit learning outcomes: a. describes the fundamental principles of object-oriented programming; b. interprets a user's needs while dealing with simple program specifications; c. design a simple class using UML notation; d. creates a simple application based on UML design and the incremental development process of coding, debugging, and testing; e. applies basic control structures – sequence, repetition, and selection – to program development; f. produces simple interactive applications.
Weight	**30%**
Total Marks	50 marks
Word limit	Not applicable
Due Date	**Week 11 (28/09/2022)**
Submission Guidelines	• All work must be submitted on Moodle by the due date along with: • A completed Assignment Cover Page • Word document with all tasks and program output's screenshots (MS Word format: 1.5 spacing, 11-pt Calibri (Body)font and 2 cm margins on all four sides of your page with appropriate sectionheadings) • All Python code files • Reference sources must be cited in the text of the report, and listed appropriately at the end in a reference list using IEEE referencing style.
Extension	If an extension of time to submit work is required, a Special Consideration Application must be submitted directly on AMS. You must submit this application three working days prior to the due date of the assignment. Further information is available at: https://www.mit.edu.au/about-us/governance/institute-rules-policies-and-plans/policies-procedures-and- guidelines/assessment-policy
Academic Misconduct	• Academic Misconduct is a serious offence. Depending on the seriousness ofthe case, penalties can vary from a written warning or zero marks to exclusion from the course or rescinding the degree. Students should make themselves familiar with the full policy and procedure Available at: https://www.mit.edu.au/about-mit/institute-publications/policies-procedures-and-guidelines/AcademicIntegrityPolicyAndProcedure For further information, please refer to the Academic IntegritySection in your Unit Description.

Assignment Description

This assignment will give you practical exposure with interactive programs, conditional execution (if/else statements, for and while loops) methods that return values, text processing, python naming convention, use Module with import and simple class implementation.

You will be developing a simple and easy to play "Mastermind" game. Mastermind is a two-player code-breaking game, in which one player hides a code consisting of colors, while the other player has to guess it using clues given by the former player for each turn.

In this assignment, you will be creating your own Mastermind Game, a simple version, using Python Language. In your version of Mastermind, the program will ***randomly*** select a three (3) digits secret code and the user tries to guess it, based on the deterministic clues given by the program. You must guess a secret three-digit number based on clues.

The game offers one of the following hints in response to your guess: "Yellow" when your guess has a correct digit in the wrong place, "Green" when your guess has a correct digit in the correct place, and "Red" if your guess has no correct digits, and ***it will not tell you which position it got right***. You have 10 tries to guess the secret number.

After a correct guess by the user, the program prompts- if the user wants to play the game again, if the user chooses not **(No)** to play again, then program displays the various statistics about the series of games played by the user as shown in the example log of execution below. **Your program output must be exactly as the example log provided.**

Example Log of execution (user input underlined)

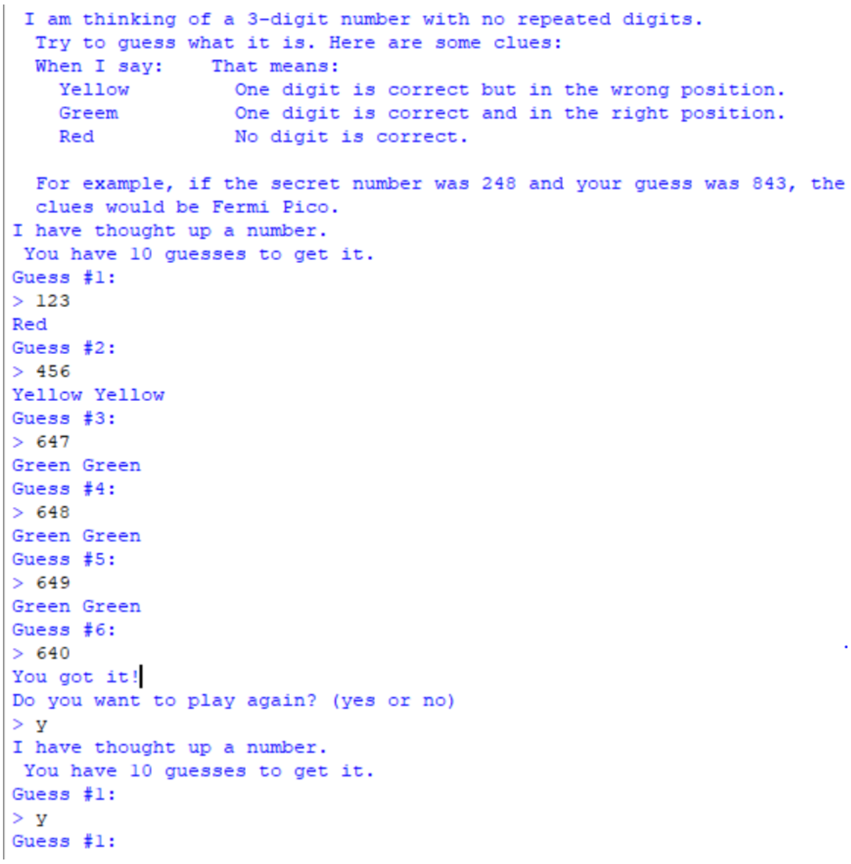
```
          ~~~ Welcome to Mastermind Game ~~~
     Hi gamer, welcome to the Mastermind game
     What is your name? Harry
     Hello, Harry, Please follow the given instructions to play the game.

           ~~~GAME INSTRUCTION~~~
   I am thinking of a 3-digit number. Try to guess what it is.
   Here are some clues:
   When I say:    That means:
   Yellow         One digit is correct but in the wrong position.
   Green          One digit is correct and in the right position.
   Red            No digit is correct.

   For example, if the secret number was 346 and your guess was 843, the
   clues would be Green Yellow.'''
```

Prompt >	I have thought up a number. You have 10 guesses to get it. Guess #1:
Player input>	457

Prompt >	Yellow Guess #2:
Player input>	789
Prompt >	Red Guess #3:
Player input>	563
Prompt >	Yellow Guess #4:
Player input>	215
Prompt >	Red Guess #5:
Player input>	340
Prompt >	Green Yellow Yellow Guess #6:
Player input>	043
Prompt >	Yellow Yellow Yellow Guess #7:
Player input>	304
Prompt >	You got it! Do you want to play again? (yes or no)
Player input>	Yes
Prompt >	I have thought up a number. You have 10 guesses to get it. Guess #1:
Player input>	120
Prompt >	Red Guess #2:
Player input>	345
Prompt >	Green Green Guess #3:
Player input>	346
Prompt >	Green Guess #4:
Player input>	357
Prompt >	Green Yellow Guess #5:

Player input>	385
Prompt >	Green Green Guess #6:
Player input>	365
Prompt >	Green Green Guess #7:
Player input>	395
Prompt >	You got it right in 7 guesses! Do you want to play again? (yes or no)
Player input>	No
Display>	Thank you for playing Overall results: Total games = 2 Total guesses = 15 Average guesses/game = 7.5

A sample screenshot of the game:



```
 I am thinking of a 3-digit number with no repeated digits.
  Try to guess what it is. Here are some clues:
  When I say:    That means:
    Yellow         One digit is correct but in the wrong position.
    Greem          One digit is correct and in the right position.
    Red            No digit is correct.

  For example, if the secret number was 248 and your guess was 843, the
  clues would be Fermi Pico.
I have thought up a number.
 You have 10 guesses to get it.
Guess #1:
> 123
Red
Guess #2:
> 456
Yellow Yellow
Guess #3:
> 647
Green Green
Guess #4:
> 648
Green Green
Guess #5:
> 649
Green Green
Guess #6:
> 640
You got it!
Do you want to play again? (yes or no)
> y
I have thought up a number.
 You have 10 guesses to get it.
Guess #1:
> y
Guess #1:
```

Tasks to be completed for Assignment-2

1. Draw a UML notation for class diagram, class name “MasterMind”, showing class name, all attributes, constructor and methods (this class you need to define for your program).

2. Define two (2) class constants for the maximum_number guesses (10) and number_of_digits (3) to guess

3. Correctly implement the following in addition to method main:

   a) Create a Module “gameContent.py”

   b) Import this module to “functionGame.py”, similar to this:
   from gameContent import MasterMind

   c) You can develop your code as you like but you must have these following methods:

      - disp (------) to display initial instructions to play the game

      - guessCode(…….) which will Returns a string made up of number_of_digits (3) unique random digits.

        # Create a list of digits 0 to 9.
        # generates a random number and Shuffle them into random order.

      - guessFlagColor(…….) which will return a string with the yellow, green, red clues for a guess and secret number.

   d) dispResult(……) which will display game results. At the end of the log you should print a summary:
   # About the series of games played by the user,these being the total number of games played (use a variable to keep track of this)
   #The total number of guesses made (use another variable to keep track of this)
   #The average number of guesses per game.

4. When you ask the user whether or not to play again, you need to consider following:
   - You should continue playing if this answer begins with the letter “y” or the letter “Y”. Notice that the user can type words like “yes”. You must look just at the first letter of the user’s response and see whether it begins with a “y” or “n” (either capitalized or not) to determine whether to play again or not.

Marking criteria: (Refer to “Marking Guide” also on the next page)

Program Demonstration: Students must demonstrate their program in week12 (lab time) to the tutor and are expected to explain their implementation. Marks will be deducted (50% -maximum) for no demonstration or poor explanation.

Marking criteria is shown in following table. Marks are allocated as follows:	
Description of criteria	**Marks**
Task 1	
Comments describing program, author, date are included and	0.5
UML diagram of Mastermind class	1
Import statement is included	0.5
Task 2	
Class constant is declared and initialised	1
Define and proper use of Class and Module	1.5
Task 3	
Correctly calls method that generates random number	1
Method to print instructions is correctly declared	2
Method to play game is implemented and runs successfully	10
Task 4	
Implements a loop to allow the game to continue running until the user quits	5
Successfully uses the “ if-else” loop statements to check if user guess is correct, so break out of this loop.	5
Successfully uses the “ if-else” loop statements to check if *number of Guesses > max number of guess (10)*, so print('You ran out of guesses.')	5
Task 5	
Method to report overall results is implemented and runs successfully	5
Others	
Use whitespaces (vertical & indentation) properly to make program more readable.	0.5
Follow naming convention; give meaningful names to methods and variables in your code.	1
Localize variables whenever possible -- that is, declare them in the smallest scope in which they are needed.	1
Your program compiles successfully without any errors	5
Your program is interactive and gives correct output	5
Total marks	**50**

Marking Rubric for Assignment # *2:* Total Marks 50

Grade Mark	Excellent 80%+	Very Good 70%-79%	Good 60%-69%	Satisfactory 50%-59%	Unsatisfactory <50%
Task 1/ 0.5 + 1 +0.5	Program is well documented, includes import statements to support external classes and, UML diagram	Program is well documented but has a few minor issues	Generally, well documented and named buthas some issues	Program does not followstated requirement	This is not relevant to the assignment requirements
Task 2/ 1 + 1.5	Constant is declared appropriately, proper use of module	Constant is declared but not as per specification, module is created not as per specification	Constant is declared but has Some issues. Module is defined but has some is sues	Constant is declared butnot as per naming convention. Module is define but not imported properly	This is not relevant to the assignment requirements.
Task 3/ 1 + 2 + 10	Method design and implementation is clear and easy-to-follow	Method design and Implementation is consistent	Method design and implementation is mostly consistent	Method design and implementation is adequate but misses the requirements	No methods or method implementation is incorrect
Task 4/ 5 + 5 + 5	Loop is correctly and successfully used following all requirements exactly	Loop is Implemented correctly and works, following most of the requirements.	Part of the loop is implemented and works but does not meet all the requirements	Most components present	No loop used, or implementation is incorrect
Task 5/ 5	Method to output resultsis well implemented and runs successfully	Method to output results is mostly well implemented	Good effort made but not outstanding	Made some effort.	Output is incorrect, or method not implemented
Other Tasks/ 0.5 + 1 + 1 + 5 + 5	Follows syntax rules and Python programming conventions, program runs correctly	Very good effort is made in following Python syntax and conventions	Good effort made but not outstanding	Made some effort	Program completely failsto compile and run