code.sprint MT

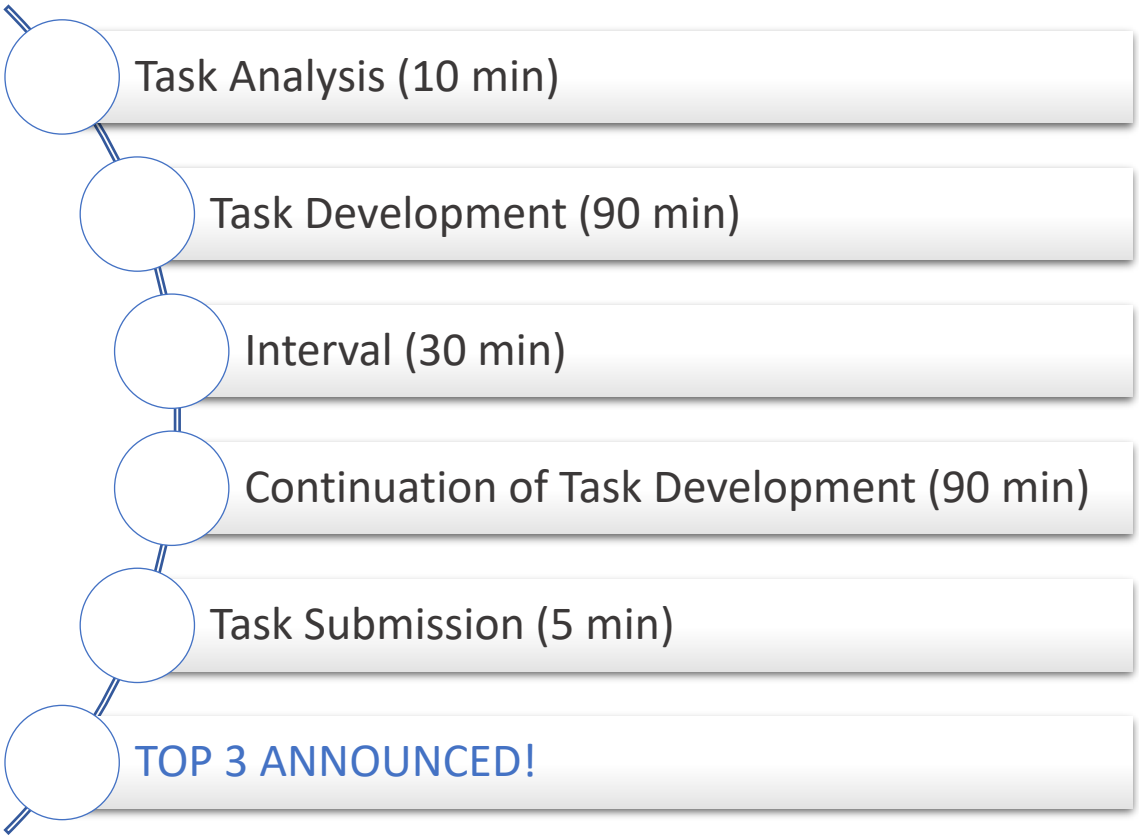
TASK BOOKLET

- Final Round -

SEC 2020

DIRECTORATE FOR LEARNING &
ASSESSMENT PROGRAMMES

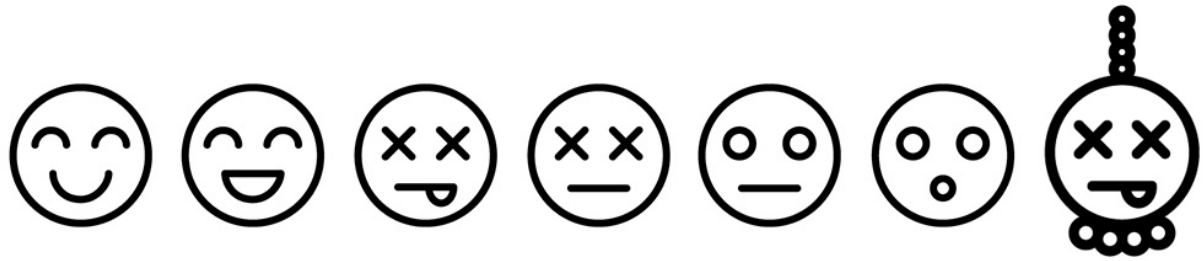
Final Round Schedule



Hangman (180 minutes)

Hangman is an artefact of the 19th century when criminals received the ultimate penalty for committing the ultimate crime. It's used nowadays to make learning words fun and to help people get to grips with a new language.

The purpose of the game is to guess a word by inserting letters into a series of blanks. With each incorrect guess, a new piece of the 'hangman' picture is drawn. The game is over when either the word has been guessed or the drawing is complete.



Develop a hangman according to the following functionalities:

Functionality #1 – Game Start

1. The game is stocked up with several words and a hint for each word, as per table 1.
2. A menu is shown with four options:
   - 1) Easy
   - 2) Medium
   - 3) Hard
   - 4) Exit

Functionality #2 – Gameplay

1. The game starts with eight (8) lives.
2. A word is chosen at random from table 1 according to the level of difficulty.
3. A series of underscores '_' are shown for each letter of the word chosen.
4. The players must repeatedly input a letter until the word is fully guessed or all lives have been used.
5. For each incorrect letter, a life is deducted, and the hangman is created as in table 2. For example, if the player enters the letter 'h', and it is not part of the word selected, one life is lost.
6. The player can choose to get a hint by inputting '?'. A hint is given against a number of lives as follows:
   - Hint for Easy deducts 2 lives,
   - Hint for Medium deducts 3 lives,
   - Hint for Hard deducts 4 lives.

7. At the end of the game, if the player:
    - loses all lives, the word selected is shown,
    - guesses the word, the number of turns taken to guess the word is displayed.

** A sample screen shot is shown in Figure 1.*

Validation

- Menu option requires the user to enter 1, 2, 3 and 4 only. Avoid runtime error when the player enters characters instead on numbers.
- During gameplay, players cannot enter a letter, if it has been already chosen.
- The word selected is not case sensitive.

** Proper messages are displayed if the input does not abide by the validation rules.*

User Interface

- The interface should display a representation of the hangman according to the lives remaining, as in Table 2.
- After every turn there should be an indication of the available remaining letters that the user can input.
- If the user enters a letter which is part of the word chosen, the underscore is replaced by the letter. For example, if the word chosen is Array, and the user enters letter 'a', the game should indicate **'A _ _ A _'**.

Game Addon

1. The player can either input a letter or automatically tries to guess the entire word.
2. If the player enters an entire word that contains some letters which are included in the word randomised, the respective underscores are replaced by the letters. For example, if the word randomised is 'arrays', and the user enters the word 'Larries', the game should display **'A R R A _ S'** and three lives are lost.

```
Lives Remaining: 3
  --------|
      |   |
      _   |
     |_|  |
          |
          |
          |
  --------|

Available Letters:
- B C - - - G H I J - L M - O P - R S - U V W - Y Z

Guessed Word:
_ d _ _ a t _ _ n

>> Enter letter:
```

Figure 1: Sample screenshot

Word	Hint	Category
Ethics	Deciding what is morally right and wrong.	Easy
Google	This knows everything!	Easy
Privacy	Keeping yourself and your personal data safe when off and online.	Easy
Algorithm	A clear design of how to solve a problem.	Medium
Creativity	The use of imagination or original ideas to create something.	Medium
Flowchart	A symbolic diagram to a problem solution.	Medium
Computational	'Why' and 'How to' are the essential foci of this type of thinking.	Hard
Decomposition	The process of breaking a larger problem down into smaller modular components.	Hard
Generalisation	Adapting solutions to be used to solve a range of problems.	Hard

Table 1: List of words

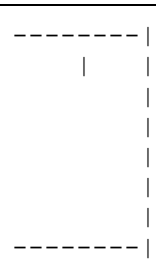
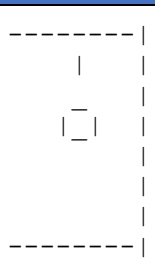
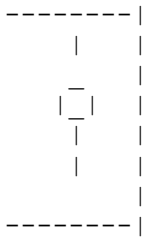
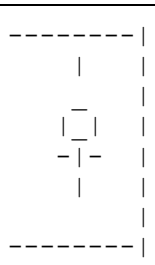
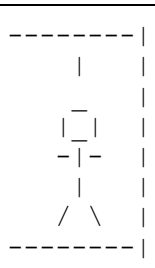
Lives	Graphic
7	
6	
5	
4	
3	
2	
1	
Game Over	

Table 2: Graphics representing number of lives

Name the class containing the main method **RunApp.**

Submit your program in a folder called **Hangman**

Hints

1. To ignore case sensitivity of the letters, the player's input can be changed into its uppercase equivalent. For example, if the player enters character 'j', it can be changed to 'J'. This can be done using the code:
   `playerInput = Character.toUpperCase(playerInput);`

2. To check the length of a String, the `.length()` can be used. In the example below, variable num_of_letters will hold value 5, because the word HELLO has 5 letters:

```
String name = "HELLO";
int num_of_letters = name.length();
```

3. To check individual letters in a String, the `.charAt()` can be used. In the example below, the word HELLO is split into separate characters and stored in an array of five elements (the same length as the word HELLO).

```
String name = "HELLO";
int word_length = name.length();
char[] word_in_letters = new char[word_length];
for (int i=0; i<name.length(); i++)
        word_in_letters[i] = name.charAt(i);
```

Assessment Rubric

Program Functionality	User-Friendly Interface	Proper use of Comments	Proper Conventions (Camel case, meaningful var names etc.)	Name of Folder & Class/es	User Input	Suitable Prompts / Messages displayed
Randomisation (word chosen according to difficulty)	Arithmetic Operations (number of lives)	Validation (menu options)	Validation (letters available)	Validation (no case sensitivity)	Proper use of Arrays	Game ADD ON
Code Efficiency (avoid duplicate lines of code)	Other Features (not listed in task)	**Maximum Score: 30** **+ 2 for every extra feature.**				
0 – Not Satisfactorily | 1- Partly Satisfactorily | 2- Entirely Satisfactorily						

ICE Malta

eSkills Malta Foundation

L-Università ta' Malta
Faculty of Information & Communication Technology

Middlesex University Malta

mita