

Programming in C++ 063 Project

Project Overview

Each project team is required to create a **single program to recreate the famous WORDLE game**, where you are given a fixed number of chances to guess a secret word (i.e., answer). The player will type in a word as a guess and the program will determine which letter(s) is/are in the secret word and which is/are not. The player will continue to guess the correct answer the within six chances or they will lose the game.

Each project team will be awarded a specific subject to create between 6-10 words/terms to test their user, from the following:

- Mathematics
- Physics
- Chemistry
- Biology
- Computing

Once the subject is identified for each group, they will work on the topic and create a program using C++ Programming Language which includes the following modules:

- Module 1: Let's Play (insert subject name here) WORDLE
- Module 2: Check Score List
- Module 3: (insert subject name here) Wordle Developer (insert Group's name)
- Exit Program

The details of the project are given below:

Start of the Program

1. When the user starts the program, in the first screen, the program welcomes user to the program and then requests for the user to enter the following information:
 - a. User's name (with spaces included/allowed)
 - b. A valid input/selection of Gender (for the purpose of displaying salutation (i.e., Mr. or Ms.) for the user's name).
 - c. The program will verify if the Gender entered is valid or invalid. If it is invalid, the program will inform the user and repeat the gender input process.
2. After the user's details are saved, the program will then display a **Selection Menu** with the 4 modules mentioned above, with the user's name and his/her salutation appearing at the top of the screen.

Example:

```

      W E L L C O M E
    T O S T A R W A R S
      W O R D L E

=====
~Selection Menu~
Please select from the following items, Mr. Rambo Roy
    1) Module 1: Let's Play Star Wars Wordle
    2) Module 2: Check Score List
    3) Module 3: Star Wars Wordle Developer (1E1-A)
    4) Exit Program
=====
```

Module 1: “Learn About”

1. When the user selects **Module 1**, your program will clear the console window Selection Menu screen (using system-clear function) and displays the Module Name and user’s name at the top of the console window (e.g., “*Module 1: Let’s Play Star Wars Wordle, Mr. Rambo Roy*”). Before the game begins, your program will provide a set of instructions on the rules and how to play the game.

Example:

```

LETS PLAY STAR WARS WORDLE

Welcome to Module 1: Let’s Play Star Wars Wordle, Mr. Rambo Roy!

=====
Here are the rules:
> There are 10 words in this game
> For each word, you are given five attempts to guess the word
> You may type in any characters with uppercase or lowercase
> After each guess, we will check and provide feedback on each letter entered
> For each correct answer, you will be awarded 10 points
=====

Press any key to continue...
```

2. The game will start from guessing the 1st word until the last word. There will be an indicator of which word is currently being displayed (i.e., This is Word1 of 10).
3. For each word, game begins by showing the empty spaces on the screen and ask the user to type in their guess. Once entered, your program will check:
 - a. If the letter(s) exists or does not exist in the secret word
 - b. If the letter exists, inform the player if they are in the correct position or not in the secret word
 - c. If all the letters are in the correct position, program will congratulate the user, clears the screen and move on to the next word.
4. After each word has been completed, if the user managed to guess it correctly, the total score for the user will be updated. If the player did not manage to guess it right, no changes are made to the total score. The current score of the user will be displayed after each round/word is completed.

Example (red indicates the user’s input):

```

This is Star Wars Word1 of 10 (Hint: Famous transportation name in Star Wars)

[ ][ ][ ][ ][ ]

Enter your Guess1 for Word1: master

[ S ][ P ][ A ][ C ][ E ]

For your Guess1:
S > is not inside Word1
P > is not inside Word1
A > is not inside Word1
C > is not inside Word1
```

Programming in C++ 063 Project

E > is not inside Word1

Enter your Guess2 for Word1: **venus**

V	E	N	U	S
---	---	---	---	---

For your Guess2:

V > is not inside Word1

E > is not inside Word1

N > is inside Word1, but it's in the wrong position

U > is not inside Word1

S > is not inside Word1

Enter your Guess3 for Word1: **wring**

W	R	I	N	G
---	---	---	---	---

For your Guess3:

W > is inside Word1, but it's in the wrong position

R > is not inside Word1

I > is inside Word1 and it's in the right position

N > is inside Word1 and it's in the right position

G > is inside Word1 and it's in the right position

Enter your Guess4 for Word1: **xwing**

X	W	I	N	G
---	---	---	---	---

For your Guess4:

X > is inside Word1 and it's in the right position

W > is inside Word1 and it's in the right position

I > is inside Word1 and it's in the right position

N > is inside Word1 and it's in the right position

G > is inside Word1 and it's in the right position

Congratulations! You have guessed Word1 right!

Total Score: 10

Press any key to continue...

5. Once the user pressed any key to continue, all the guesses from Word1 will be cleared and program starts the game for Word2.

This is Star Wars Word2 of 10 (Hint: One of the androids in Star Wars)

--	--	--	--	--

Enter your Guess1 for Word2: **bb-89**

B	B	-	8	9
---	---	---	---	---

For your Guess1:

B > is not inside Word2
B > is not inside Word2
- > is inside Word2, but it's in the wrong position
8 > is not inside Word2
9 > is not inside Word2

Enter your Guess2 for Word2: **c-3P0**

C	-	3	P	0
---	---	---	---	---

For your Guess2:

C > is not inside Word2
- > is inside Word2 and it's in the right position
3 > is not inside Word2
P > is not inside Word2
0 > is inside Word2 and it's in the right position

Enter your Guess3 for Word2: **R-2D0**

R	-	2	D	0
---	---	---	---	---

For your Guess3:

R > is not inside Word2
- > is inside Word2 and it's in the right position
2 > is inside Word2 and it's in the right position
D > is not inside Word2
0 > is inside Word2 and it's in the right position

Enter your Guess4 for Word2: **n-2t0**

N	-	2	T	0
---	---	---	---	---

For your Guess4:

N > is not inside Word2
- > is inside Word2 and it's in the right position
2 > is inside Word2 and it's in the right position
T > is not inside Word2
0 > is inside Word2 and it's in the right position

Enter your Guess5 for Word2: **S-2V0**

S	-	2	V	0
---	---	---	---	---

For your Guess5:

H > is inside Word2, but it's in the wrong position

Programming in C++ 063 Project

```
- > is inside Word2 and it's in the right position
2 > is inside Word2 and it's in the right position
T > is not inside Word2
0 > is inside Word2 and it's in the right position
```

Sorry! You did not guess Word2 right!

Total Score: 10

Press any key to continue...

6. At the end of the game, the user's name and total score will then be recorded into the ScoreList.txt text file using file processing, together with the previous user names and total scores of players who have attempted the Wordle game previously.
7. The user will be given an option to repeat the Wordle Game or Exit to the Selection Menu.

Enter your Guess5 for Word10: **Anakin**

A	N	A	K	I	N
---	---	---	---	---	---

For your Guess5:

```
A > is inside Word10 and it's in the right position
N > is inside Word10 and it's in the right position
A > is inside Word10 and it's in the right position
K > is inside Word10 and it's in the right position
I > is inside Word10 and it's in the right position
N > is inside Word10 and it's in the right position
```

Congratulation! You have guessed Word10 right!

Total Score: 80

Your name and score have been added into the ScoreList.txt file.

Press 1 to repeat the Wordle Game, Press 2 to go back to Selection Menu

Module 2: "Check Score List"

1. When the user chooses **Module 2**, your program will clear the console window Selection Menu screen (using system-clear function) and displays the Module Name and user's name at the top of the console window.
2. The program will then extract the information from ScoreList.txt and show the list of user names and total scores obtained from Module1.

Example:

```

W O R D L E   S C O R E   L I S T

```

Welcome to Module 2: Check Score List, Mr. Rambo Roy!

Here is score list:

User Name	Score
=====	
Rambo Roy	80
Rambo Roy	50
Big Bang	90
Small Bang	70

Press any key to continue...

3. Once the user pressed any key to continue, the program returns to the Selection Menu.

Module 3: "Developer Information"

1. When user selects **Module 3**, your program will clear the console window Selection Menu screen (using system-clear function) and displays the Module Name and user's name at the top of the console window.
2. The program then opens a group photo of the project team. The photo must include the team member's name and student ID.
3. Then, the program proceeds to provided description on each of the team members, complete with team member's names, student IDs and job/task allocation for the project.
4. At the end of the Module, the program allows clears the screen and exit to the Selection Menu.

Example:

```

(      )      (      )      (      )      (      )      (      )      (      )      (      )
)\    )\    )\    )\    )\    )\    )\    )\    )\    )\    )\    )\    )\    )\    )\    )\
((/((  ((/((  ((/((  ((/((  ((/((  ((/((  ((/((  ((/((  ((/((  ((/((  ((/((  ((/((  ((/((
/((_)  /((_)  /((_)  /((_)  /((_)  /((_)  /((_)  /((_)  /((_)  /((_)  /((_)  /((_)  /((_)  /((_)
(_)_  (_)_  (_)_  (_)_  (_)_  (_)_  (_)_  (_)_  (_)_  (_)_  (_)_  (_)_  (_)_  (_)_  (_)_  (_)_
|_) | |_) | |_) | |_) | |_) | |_) | |_) | |_) | |_) | |_) | |_) | |_) | |_) | |_) | |_) | |_)
|_) | |_) | |_) | |_) | |_) | |_) | |_) | |_) | |_) | |_) | |_) | |_) | |_) | |_) | |_) | |_)

```

Welcome to Module3: Star Wars Wordle Developer Information, Mr. Rambo Roy

Team Member	Task/Job Description
=====	
Papa Roy	Program and Design Module 1 word checking and testing
Mama Roy	Research on Word List and assist in testing Module1
Sister Roy	Create and integrate other modules with Selection Menu
Brother Roy	Debugging of Errors and Integration of all modules
Small Baby Roy	Program and debug Module 2 and Module 3

Press any key to continue...

Module 4: Exit Program

For the **Exit Program** selection menu, the program must follow the following requirements:

- a. If user has not accessed or have not completed Module 1, the program will display the following question: ***“Are you sure you want to exit before playing Star Wars Wordle, Mr. Rambo Roy?”***
 - i. If user answers **Y: Yes**; program exits and end its execution.
 - ii. If user answers **N: No**; program will return to the **Selection Menu**.
- b. If user has completed Module 1, the program will display the following question: ***“Are you sure you want to exit the program, Mr. Rambo Roy?”***
 - i. If user answers **Y: Yes**, program will display the following message ***“Thank you Mr. Rambo Roy for playing Star Wars Wordle!”***
 - ii. If user answers **N: No**; program will return to the **Selection Menu**.
- c. Any invalid feedback from user will prompt an error message until user answer with the correct options provided.

Your project is expected to showcase proper usage of:

- a. Loop constructs without any logical error or premature exit/end.
- b. Decision constructs and proper notification/prompt to the user.
- c. At least 1 class.
- d. At least one constructor.
- e. Passing parameters/arguments using Functions/Member Functions. No usage of global variable is allowed.
- f. Module 1 and Module 2 MUST be separated into individual User-defined Functions or Member Functions (from a class/classes).
- d. All the coding in the source code must be properly indented and readable.
- e. For the final submission, your source code must not contain any comment lines or programming descriptive notes.
- f. No hard copy or printed version of the project is required. Final project code will be submitted via Google Drive (will be informed later)

Project Marking

This project is worth 30% of the unit marks. The marks breakdown is given below:

- Project Code (**60 Marks converted into 20%**)
- Project Interview Marks (**5%**)
- Self and Peer Assessment (**5%**)

Important Points & Reminders

1. A project team should comprise of 5-6 members and each team will submit ONE single project.
2. A backup copy of the project should be kept in case the project is lost in any way. It is the team's responsibility to ensure that backup copies are kept and the copies are timely updated.
3. If the project of any team is found to be copied from another group, both teams will be penalized and severe action would be taken as listed in the Unit Outline. It would be wise for the team to work in close proximity with each team members and refrain from sharing their project code with other teams.
4. Every group member will be awarded with the SAME marks for the Project Code section since this is a group project and the group submit ONE single project. **However, if there is/are issues reported on any of the team member's failure to contribute significant amount of work**

towards completing the project, the team member will only be awarded the individual mark (10%) (Presentation and Self/Peer Assessment) out of the overall 30%.

Project Interview

1. During the Project Interview session, the team will submit the final project code in soft copy format and submit it to their respective lecturer during their designated Presentation Slot (in week 16).
2. Each member of the team would be asked to answer a series of questions relating to the project establishing their understanding and contribution to the project. Each member may be asked on the part that they were heavily involved in or any other parts of the project.
3. Each member of the group would be given Self & Peer Assessment form in which they will have to assess honestly on individual contribution of their team members and themselves. It is possible for a team member to get **zero marks** for this section of the project after going through the Self & Peer Assessment process.
4. Any member of the team who failed to attend the presentation without valid reason will not be awarded any marks for the Presentation Marks (5%) and Self and Peer Assessment (5%).

Project Submission

The project is to be submitted on **Week 16**, as stated in the Unit Outline. Please book the presentation slot for your team once the unit lecturer has made it available to you. Each group will only be allowed to book ONE slot for their Presentation Session and all team members must be available for the session. The project code submitted at the end of the presentation session will be considered as the final copy of the project and no further changes to the code is allowed.

Programming in C++ 063 Project

Programming in C++ 063 Project Marking Guide

Item	Description	Marks breakdown	Checklist
1.	Meeting Presentation Marks (Individual Marks)	5%	
2.	Peer Assessment (Individual Marks)	5%	
3.	Project Code (Team Marks): 60 marks converted into 20%		
	Basic Requirements	18 marks in total	
	1. Source code are easy to read and aligned <ul style="list-style-type: none"> a. Readable with No comment lines (1) b. Aligned properly (2) 	3 marks	
	2. Correct usage of Loop constructs <ul style="list-style-type: none"> a. No infinite loop (2) b. Effective use of for, while or do while loop (2) 	4 marks	
	3. Correct usage of Class/Functions <ul style="list-style-type: none"> a. At least 1 class (1) b. Class is called/used effectively in the program via object(s) (2) c. Included at least 1 constructor (1) 	4 marks	
	4. Correct use of Decision constructs <ul style="list-style-type: none"> a. Effective use of if, if else or switch case (2) b. No errors occurred in using constructs (2) 	4 marks	
	5. Passing values with parameters only. (No-global variable is used.)	3 marks	
	Project Interface Requirements:	10 marks in total	
	1. Module Name, User's name and salutation <ul style="list-style-type: none"> a. User name + Salutation appears in all the Modules (2) b. Module Name clearly displayed its own modules/sections (2) 	4 marks	
	2. Program exits only from Selection Menu (Option 4: Exit Program)	2 marks	
	3. Appropriate feedbacks/responses are given to the user throughout all the modules and selection menu	2 marks	
	4. Screen is cleared and module content appears at the top of console window	2 marks	
	Module Specifications:	25 marks in total	
	1. "Module 1: Let's Play Wordle": <ul style="list-style-type: none"> a. Program able to check word using uppercase or lowercase format (3) b. Program provide correct feedback for each letter from the guess word entered (3) c. Word count is provided (i.e., Word2 of 10, Word2 of 6) at the start of each new word (1) d. Each new word is displayed on a new/cleared screen to start a guess (1) e. Correct total marks provided after each guess completion (3) f. Provided an option to repeat the module (2) g. Provided an option to exit to Selection Menu (1) 	14 marks	
	2. "Module 2: Check Score List": <ul style="list-style-type: none"> a. Program saved all record of previous attempt from Module 2 into ScoreList.txt file (2) b. Program shows each user's names and scores correctly (2) 	4 marks	
	3. "Module 3: Developer Information": <ul style="list-style-type: none"> a. Successfully display photo with name and student ID of all team member (2) b. Clearly show accurate description on each team members task/responsibilities (2) 	4 marks	

	<p>4. Exit Module: Program prompts the user if he/she has not accessed Module 1 before exiting</p> <ul style="list-style-type: none">a. Check for previous access to Module 1 (2)b. Provide option to continue or exit the program with salutation (1)	3 marks	
	<p>Overall creativity in designing the project and additional/unique features, which may include (but not limited to):</p> <ul style="list-style-type: none">a. Menu/Selection design is consistent in all the modulesb. Included additional design that supports the theme of the topic chosenc. Animation or Special Text Effects includedd. Changing of text colors and background of console window to suit the theme of the topic chosene. Effective use of additional function(s) that is/are not covered in the syllabusf. Text and content are consistently aligned across all the modulesg. Any other features deemed appropriate and suitable to support the project requirements	Maximum of 7 marks	

Programming in C++ Team Project - Presentation Marking Rubric (Individual)

Each student will be asked to elaborate on ONE/TWO basic requirements from ANY parts of the project specification/requirements. The details provided in the explanation will be marked based on the following rubric:

<u>Name of Student:</u>		<u>Total Marks:</u>	
	1	0.5	0
Subject knowledge	Demonstrate full knowledge on the topic/part questioned	Demonstrate adequate knowledge on the topic/part	No knowledge demonstrated on the topic/part questioned
Elaboration/Explanation	Provide clear explanation on the question	Provide sufficient explanation on the question	Provides poor/no explanation on the question
Program code reference	Explanation includes accurate reference/evidence from the program code	Explanation includes fairly accurate reference/evidence from the project code	Explanation fails to include reference/evidence from the program code
Delivery	Uses correct jargon(s) and terminology in answering	Included poor usage jargon(s) and terminology in answering	Incorrect jargon(s) and terminology used in answering
Overall Quality of Presentation	Able to provide a full demonstration on answer and explanation in a convincing and effective manner	Able to provide an adequate demonstration on answer	Failed to provide a satisfactory and convincing demonstration on answer

Programming in C++ Team Project - Self and Peer Evaluation Form (Individual)

Categories	1 point	0.5 point	0 point
A) Contributions	Routinely provides useful ideas when participating in the group and in classroom discussion. <u>A leader who contributes a lot of effort.</u>	Sometimes provides useful ideas when participating in the group. <u>A satisfactory group member who does what is required.</u>	Rarely provides useful ideas when participating in the group and in classroom discussion. <u>May refuse to participate.</u>
B) Problem-solving	Actively looks for and suggests solutions to problems.	Does not suggest or refine solutions but is willing to try out solutions suggested by others.	Does not try to solve problems or help others solve problems and lets others do the work.
C) Attitude	Is never publicly critical of the project or the work of others. <u>Always has a positive attitude about the task(s).</u>	Is occasionally publicly critical of the project or the work of other members of the group. <u>Usually has a positive attitude about the task(s).</u>	Is often publicly critical of the project or the work of other members of the group. <u>Is often negative about the task(s).</u>
D) Focus on the task	Consistently stays focused on the task and what needs to be done. <u>Very self-directed.</u>	Focuses on the task and what needs to be done some of the time. Needs to be reminded sometimes.	Rarely focuses on the task and what needs to be done. <u>Let others do the work.</u>
E) Teamwork	Almost always listens to, shares with, and supports the efforts of others. Tries to keep people working well together.	Often listens to, shares with, and supports the efforts of others, but sometimes is not a good team member.	Rarely listens to, shares with, and supports the efforts of others. Often is not a good team player.

Please evaluate your team members and yourself using the guideline provided above

Team Members' Names	A) Contributions	B) Problem-Solving	C) Attitude	D) Focus	E) Teamwork	Total Points
1)						
2)						
3)						
4)						
5)						
6)						