Procedures & Functions: Guess the Word

# Procedures

|  |  |
| --- | --- |
| **What is a procedure? (Describe)** | |
|  | |
|  | **Describe what the example procedure on the left will do** |
| **Add 2 numbers then print the result out** |
| **What are the variables in the brackets for?** |
| **They represent numbers that will be added** |
| **Explain why procedures are useful when programming?** | |
|  | |

# Functions

|  |  |
| --- | --- |
| **What is a function? (Describe)** | |
|  | |
|  | **Describe what the example procedure on the left will do** |
|  |
| **What are the variables in the brackets for?** |
|  |
| **Explain why functions are useful when programming?** | |
|  | |

# Program Requirements

You have been asked to develop a game called Guess The Word (Hangman Style Game). This guide will show you how to develop the basic requirements of the game (Bronze)…ish, it will need tweaking slightly . There are also more extensive requirements at the end of this document (Silver and Gold)

|  |
| --- |
| **Basic Requirements**   1. There should be a pre-defined list of words 2. The game should choose a random word from the list 3. The game should the word to the user as placeholders   e.g. “STRING” would display as “\_ \_ \_ \_ \_ \_”   1. The player should be able to guess a letter 2. If the letter is correct the placeholders should be replaced with the correctly guessed letters   e.g. If “S” was guessed the following should be displayed “S \_ \_ \_ \_ \_”   1. Once all letters have been guessed correctly an appropriate “Win” message should be displayed |

# Program Design

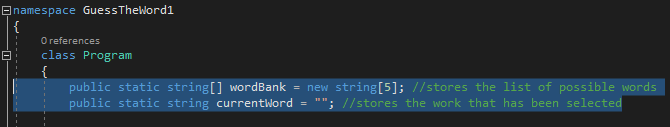
DisplayWord and GuessLetter should be repeated until all the letters in the word have been guessed correctly. Once the word has been guessed a win message will be displayed

# Programming Guide

### Global Variables

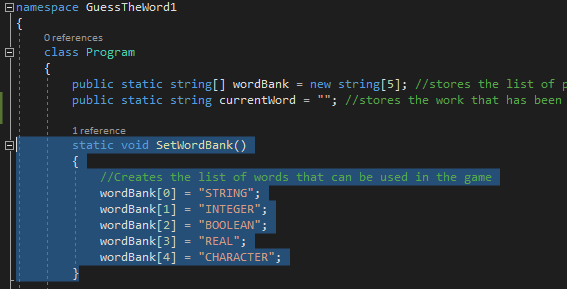
The program is going to make use of two global variables – Global variables are declared outside of any specific procedures and functions (and outside of the Main program). This means the can be “accessed” by every part of the program (can be accessed by the Main program and all the procedures and functions in the program)

**Set the following global variables at the top of your program making sure it is inside the class Program braces:**



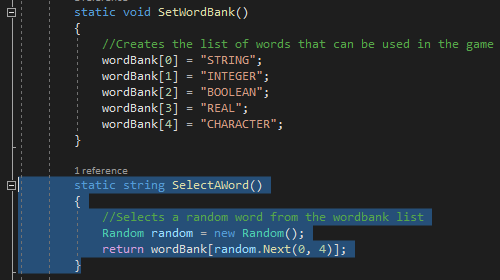
## Procedure: SetWordBank

The program needs a procedure to set up the list of words that can be chosen by the program. Note that the number of words is limited by the size of the wordBank array that you declared as a global variable (above)



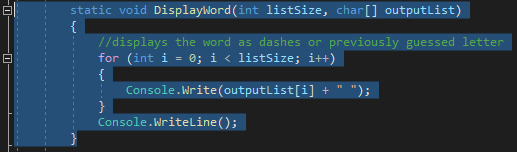
## Function: SelectWord

Next you need to add a function that can be used to select a random word from the list and return it to the main program



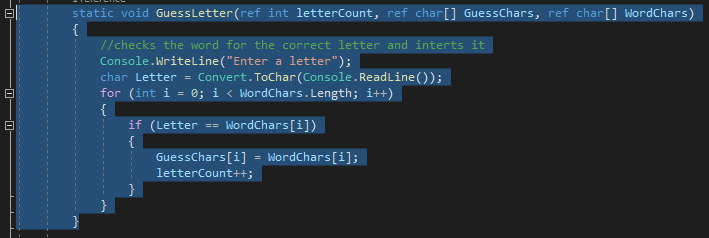
## Procedure: DisplayWord

The display word procedure is used to output the word places and any correctly guessed letters. Add the following procedure to your program



## Procedure: GuessLetter

The final procedure you need to add is to allow the user to guess a letter and update the word places if a letter is guessed correctly. Add the following procedure to your program



Now all the procedures are in place you are ready to write the Main program which is on the next page

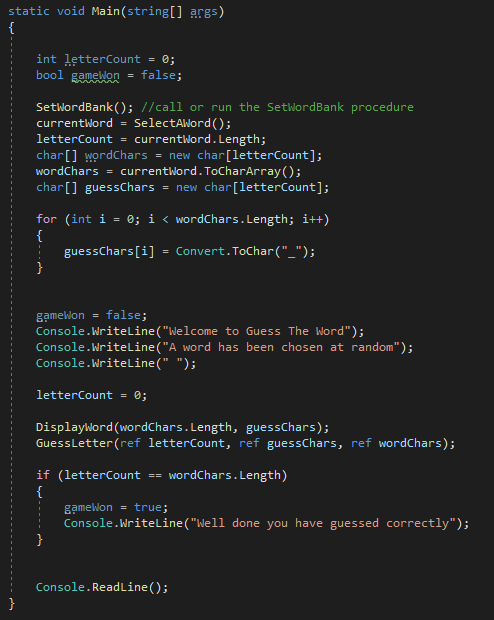
## Main Program Code

The main program needs to do the following thing

1. Declare some variables to enable the game to work
2. Call the SetWordBank
3. Store a currentWord by calling SelectAWord
4. Convert the word in to placesholders e.g. \_ \_ \_ \_
5. Display a welcome message
6. Call DisplayWord and GuessLetter to enable the game to play
7. Include a condition to display an appropriate message if the game has been won
8. **Repeat steps 5-7 until the game has been won – this does not happen in the code below**

Program your main program by following the code below

**Amend your program using a do-while loop to complete point 8 above**



# Objectives/Self-Assessment Checklists

|  |
| --- |
| **Basic Requirements**   1. There should be a pre-defined list of words 2. The game should choose a random word from the list 3. The game should the word to the user as placeholders   e.g. “STRING” would display as “\_ \_ \_ \_ \_ \_”   1. The player should be able to guess a letter 2. If the letter is correct the placeholders should be replaced with the correctly guessed letters   e.g. If “S” was guessed the following should be displayed “S \_ \_ \_ \_ \_”   1. Once all letters have been guessed correctly an appropriate “Win” message should be displayed |

|  |
| --- |
| **Intermediate Requirements**   1. Comments should be used throughout the program to describe what the code does 2. Meaningful names should be used on variables and other named objects 3. There should be a menu when the game starts with the following options    1. There should be an easy mode (This should start the game mode you have created for the Bronze/Basic requirements)    2. There should be an option to play an Intermediate mode (for now this should just exit the game)    3. There should be an option to play a Difficult mode (for now this should just exit the game) |

|  |
| --- |
| **Advanced Requirements**   1. The program should have 3 game options    1. Easy – the game is played with unlimited lives (no change)    2. Intermediate – the player has 5 live, a life is lost each time a guess is incorrect, the game is lost when the player loses all lives    3. Difficult – same as medium but with only 3 lives |

|  |
| --- |
| **Platinum Requirements**   1. Think of your own improvements/extra features. List them below and try to program them |