COMPSCI 101 Assignment – World Capitals Quiz

Due: 11:59pm, Thursday 16th October 2025.

Worth: This assignment is marked out of 60 and is worth 6% of your final mark.

Topics covered:

- Using variables
- Printing output
- Manipulating string objects
- Manipulating list objects
- Getting user input
- Using loops
- Defining functions
- Reading text files
- Writing to text files
- Using dictionaries

Submit the file containing your assignment using the Assignment Dropbox:

https://adb.auckland.ac.nz/Home/

INTRODUCTION

For this assignment you will be implementing a simple quiz, where the player can test their knowledge of capital cities around the world. A game consists of six rounds. In each round:

- The player is asked to specify the capital city of a country. They are provided with 5 options to choose from, only 1 of which is correct.
- The player has 3 opportunities to specify the correct capital city.
    - If they enter the correct capital city on their first attempt, they get 3 points for the round.
    - If they enter the correct capital city on their second attempt, they get 2 points for the round.
    - If they enter the correct capital city on the third and final attempt, they get 1 point for the round.
    - Otherwise, they get 0 points for the round.

Your program will need to:

- Display a banner.
- Run the quiz. This will involve six rounds. In each round you need to:
    - Get the data required for the question used in the round. This data will consist of the target country (the country whose capital you need to specify) and a list of 5 potential capitals (only one of which is the correct capital). **A function called `get_question_data()` is provided that does this for you – you just need to call it**.
    - Give the player 3 attempts to specify the correct capital city. The score allocated to the round is dependent on how many attempts the player requires to specify the correct capital city.
    - Add the round score to the overall score.
- Update the player's top five high scores with their current game score if necessary.

You will be provided with a skeleton file for the assignment – `Project_Skeleton.py`. The file contains:

- The statement `import random`. This statement is required to generate the data for the question used in a round. Do not remove this statement!
- An implementation of the `get_question_data()` function. This function generates and returns the data required for a question. This is a tuple consisting of a string specifying the target country, and a list of 5 strings specifying capital cities. Only one of these capital cities will be the correct option – i.e., the capital city of the target country. Do not remove or alter this function!
- A call to the `main()` function. Do not remove this statement!
- Function headers for functions you will need to complete:
    - `print_banner()`
    - `get_world_capitals_dictionary()`
    - `get_player_answer()`
    - `run_round()`
    - `run_quiz()`
    - `read_high_scores()`
    - `update_high_scores()`
    - `handle_high_scores()`
    - `main()` – Note that part of its implementation has been provided for you.

The functions that you need to complete are described below.

`print_banner()` (2 marks)

The `print_banner()` function takes a single string parameter, `username`, that represents the player's username. It uses the player's username to generate the message `"World Capitals Quiz For {username}"`. It then prints this message inside a banner. Note that the username should be printed in **uppercase**. Also note that the length of the banner is dependent on the length of the username. Not all usernames have the same length.

For example, if the function is called passing it the username `"dazh001"`, then the function would print the following banner:

```
#####################################
#  World Capitals Quiz For DAZH001  #
#####################################
```

If the function is called passing it the username `"abc001"`, then the function would print the following banner:

```
#####################################
#  World Capitals Quiz For ABC001  #
#####################################
```

The function **must** print a blank line after the banner.

`get_world_capitals_dictionary()` (2 marks)

The `get_world_capitals_dictionary()` function takes a single string parameter `filename`, representing the name of the text file containing the data on countries and their capitals. The file will consist of one or more lines of text, where each line consists of the name of a country followed by the name of its capital city, separated by a colon.

```
Country:Capital
```

The function will read the text file and use the data obtained to create and return a dictionary object, where:

- Each key is a string representing the name of a country.
- Each value is a string representing the name of the country's capital city

`get_player_answer()` (5 marks)

The `get_player_answer()` function takes two parameters:

1. `target_country` – a string representing the name of the country for which the player has to name the capital city.
2. `cities` – a list of 5 strings representing the names of 5 capital cities. The names will be unique, with one of them naming the capital city of the country specified by the first parameter.

The function will:

- Print the text "`Choices available: `" followed by the list of 5 cities.
- Print a blank line.
- Prompt the player to enter the name of the city they think is the capital for the target country.
- While the player enters a string that is not the name of one of the 5 cities in the list, the function should inform them of this and prompt them to enter another city name.
- Remove the selected city name from the list of city names.
- Return the selected city name.

For example:

```
Choices available: ['Rome', 'Honiara', 'Beijing', 'Lusaka', 'Phnom Penh']

What is the capital city of Zambia? Belgrade
You must choose from the city choices available!
What is the capital city of Zambia? Paris
You must choose from the city choices available!
What is the capital city of Zambia? Rome
```

In this example you can see that the 5 cities in the parameter list are Rome, Honiara, Beijing, Lusaka and Phnom Penh. One of these cities, Lusaka, is the capital of the target city Zambia. The player first enters Belgrade when prompted. As this is not one of the 5 cities in the parameter list, they are informed of this and prompted again. They then enter Paris, which also is not one of the 5 cities in the parameter list. Again, they are informed of this and re-prompted. Finally, they enter Rome, which is one of the cities in the list.

The function will return the string 'Rome'. The string 'Rome' is removed from the list of 5 cities so that it becomes: `['Honiara', 'Beijing', 'Lusaka', 'Phnom Penh']`. The string 'Rome' is then returned by the function.

Please note that this function does not assess whether or not the player's answer is correct. It simply makes sure they make a valid entry choosing one of the 5 cities listed and returns their selection.

`run_round()` (8 marks)

The `run_round()` function runs a round of the quiz game. It takes 2 parameters:

1. `world_capitals_dict` – a dictionary where each key/value pair is the name of a country and its capital city respectively. The keys and values are strings.
2. `countries_tested` – a list of strings specifying countries that have already been assessed in the quiz. This list is empty when the quiz begins. After every round, the target country is appended to the list. This is done to ensure that rounds in the quiz will assess unique countries.

The `run_round()` function will:

- Call the `get_question_data()` function passing it both parameters. This function is provided for you as part of the assignment. The `get_question_data()` function will return a tuple consisting of a string that will be the target country for the round and a list of 5 strings representing capital cities, one of which is the capital city for the target country. The function will also append the target country to the `countries_tested` list to ensure it is not used in the quiz again.
- Give the player 3 opportunities to correctly identify the capital city of the target country. The function calls the `get_player_answer()` function, to get the player's answer.
    - While the player still has an opportunity to identify the capital city, if the player selects an incorrect city, the message `"Your answer is incorrect! Please try again!"` is printed, followed by a blank line.
    - When the player correctly identifies the capital city, the message `"Your answer is correct! Well done!"` is printed.
    - If the player runs out of opportunities to identify the capital city, the message `"Your answer is incorrect! Better luck next time!"` is printed.
- Return the player's score for the round. If the player correctly identifies the capital city of the target country on their first attempt they get a score of 3 for the round. If they identify the capital city on their second attempt, they get a score of 2 for the round. If they identify the capital city on their third (and final) attempt, they get a score of 1 for the round. Otherwise, they get a score of 0 for the round.

An example of the output of this function can be seen on the next page. In this example, the player fails to identify the capital city of the Philippines in 3 attempts. As such, the function will return 0 as their score for the round.

```
Choices available: ['Tbilisi', 'Oslo', 'Nay Pyi Taw', 'Manila', 'Bern']

What is the capital city of Philippines? Oslo
Your answer is incorrect! Please try again!

Choices available: ['Tbilisi', 'Nay Pyi Taw', 'Manila', 'Bern']

What is the capital city of Philippines? Bern
Your answer is incorrect! Please try again!

Choices available: ['Tbilisi', 'Nay Pyi Taw', 'Manila']

What is the capital city of Philippines? Tbilisi
Your answer is incorrect! Better luck next time!
```

`run_quiz()` (6 marks)

The `run_quiz()` function runs the quiz game. It takes a single dictionary object as a parameter called `world_capitals_dict`. Each key/value pair in this dictionary consists of the name of a country and its capital city respectively. The keys and values are strings.

The `run_quiz()` function will:

- Initialize 3 variables to store information required by the quiz:
  1. A list variable to store the names of countries tested by the quiz. This list will initially be empty.
  2. An integer variable to store the player's overall quiz score. This will be 0 initially.
  3. An integer variable to store the round number. This will be 1 initially.
- Run 6 rounds of the quiz. For each round the function will:
  - Print the round number in the format: "`Round {round number}`:" followed by a blank line.
  - Call the `run_round()` function to run a round of the quiz game. This function returns the player's score for the run which is then added to the player's overall quiz score.
  - Increment the round number by 1.
  - Print a blank line.
- Print the player's overall quiz score in the format: "`You have scored {overall_score} out of 18 for the World Capital's Quiz!`".
- Return the player's overall quiz score.

`read_high_scores()` (6 marks)

The `read_high_scores()` function takes a single string parameter, `filename`, representing the name of the text file containing the player's top five high scores for the quiz game.

The text file will consist of a header in the format "`High Scores for {username}`". This will be followed by a list of the top five high scores sorted in descending order based on their value, from highest to lowest. Each high score is found on a separate line and is numbered in the format "`x. y`" where `x` is the number and `y` is the high score. The screenshot on the following page shows an example of what the high score file could look like.

```
High Scores for dazh001
1. 18
2. 17
3. 16
4. 13
5. 12
```

The function should ignore the first line of text and read the list of high scores, returning them as a list of integer values. The values in the returned list should be in the same order as they appear in the high scores list. Given the example provided, the function should return the list `[18, 17, 16, 13, 12]`.

`update_high_scores()` (6 marks)

The `update_high_scores()` function takes 4 parameters:

1. `filename` – A string representing the name of the text file containing the player's top five high scores for the quiz game.
2. `username` – A string representing the word the player's username.
3. `high_scores` – A list of five integer values representing the player's top five high scores for the quiz game.
4. `new_score` – An integer value representing the player's current score in the quiz game.

The function should update the list of top five high scores if the player's current score is high enough. The function should then write the updated list of high scores to the text file used to store this list. Note:

- The function must write the header (in the format "`High Scores for {username}`") at the top of the text file, before the list of high scores.
- The function must maintain the appropriate order of the high scores. The high scores need to be displayed in the list in descending order based on their value, from highest to lowest.

For example, if this is the content of the high scores file:

```
High Scores for dazh001
1. 18
2. 17
3. 16
4. 13
5. 12
```

And the player scored 15 for their current quiz, then the updated high scores file would look like this:

```
High Scores for dazh001
1. 18
2. 17
3. 16
4. 15
5. 13
```

If the player's current score is not high enough to fall within the list of top five high scores, the contents of the high score file remain unchanged.

handle_high_scores() (2 marks)

The `handle_high_scores()` function takes 3 parameters:

1. `filename` – A string representing the name of the text file containing the player's top five high scores for the quiz game.
2. `username` – A string representing the player's username.
3. `new_score` – An integer value representing the player's current score in the quiz game.

The function should:

- Get the list of top five high scores from the high scores text file by calling the `read_high_scores()` function.
- Update the contents of the high scores text file with the player's current score (if required) by calling the `update_high_scores()` function.

main() (3 marks)

The `main()` function takes no parameters and is used to run the quiz game program. The first two statements, both of them initialization statements, have been provided for you. Do not alter them! Update the third statement so that you initialize the variable `username` to your username.

Complete the rest of the function so that it:

1. Prints the game banner by calling the `print_banner()` function.
2. Gets a dictionary object by calling the `get_world_capitals_dictionary()` function. Each key/value pair consists of the name of a country and its capital city respectively. The keys and values are strings.
3. Runs the quiz game and obtains the player's score by calling the `run_quiz()` function.
4. Handles updating the player's high scores by calling the `handle_high_scores()` function.

An example of the game being played is shown at the end of this document.

Coding Style (20 marks)

The style mark will be determined according to the following principles:

- A docstring should be included at the top of the source file with your username and ID number.
- Variable names are meaningful and relate to the data stored in that variable (e.g. the name accurately describes the data). Variable names should be whole words where possible.
- Function names are meaningful and describe the purpose of the function.
- Clear layout is used to distinguish between different elements, e.g. spaces between operators and operands, one blank line between functions, etc.
- Constructs minimize the complexity of the program -- expressions should be simple, flow of control is not deeply nested.
- Elements are presented consistently, e.g. the standard Python code conventions are followed.
- Functions must be used to break the program up into smaller tasks. As a guideline, functions should typically be no more than 15-20 lines of code.
- Your code must not contain any "while True" or "break" statements.
- Your code must not use global variables.
- Lines of code should not be longer than 79 characters.

Marking Rubric

Implementation	40 marks
`print_banner()`	2 marks
`get_world_capitals_dictionary()`	2 marks
`get_player_answer()`	5 marks
`run_round()`	8 marks
`run_quiz()`	6 marks
`read_high_scores()`	6 marks
`update_high_scores()`	6 marks
`handle_high_scores()`	2 marks
`main()`	3 marks
Style	**20 marks**
Docstring	2 marks
Variable Names	2 marks
Function Names	2 marks
Function Length 15 - 20 lines	2 marks
Clear Layout	2 marks
Construct Complexity	2 marks
Code Conventions Followed	2 marks
No "break" or "while True"	2 marks
No global variables	2 marks
Lines of Code < 80 characters	2 marks

EXAMPLE OF THE QUIZ GAME BEING PLAYED

```
####################################
#  World Capitals Quiz For DAZH001  #
####################################

Round 1:

Choices available: ['Mbabane', 'Bamako', 'Doha', 'Podgorica', 'Manama']

What is the capital city of Swaziland? Bamako
Your answer is incorrect! Please try again!

Choices available: ['Mbabane', 'Doha', 'Podgorica', 'Manama']

What is the capital city of Swaziland? Mbabane
Your answer is correct! Well done!

Round 2:

Choices available: ['Tripoli', 'Montevideo', 'Melekeok', 'Budapest',
'Vientiane']

What is the capital city of Palau? Melekeok
Your answer is correct! Well done!

Round 3:

Choices available: ['Tbilisi', 'Antananarivo', "Nuku'alofa", 'Dublin',
'Vilnius']

What is the capital city of Georgia? Tbilisi
Your answer is correct! Well done!

Round 4:

Choices available: ['Kampala', 'Tirana', 'Tashkent', 'Abuja', 'Asmara']

What is the capital city of Nigeria? Abuje
You must choose from the city choices available!
What is the capital city of Nigeria? Abuja
Your answer is correct! Well done!

Round 5:

Choices available: ['Buenos Aires', 'Luanda', 'New Delhi', 'Guatemala City',
'Mogadishu']

What is the capital city of Argentina? Buenos Aires
Your answer is correct! Well done!

Round 6:

Choices available: ['Wellington', 'Chisinau', 'Rome', 'Tarawa Atoll', 'Lome']

What is the capital city of Togo? Lome
Your answer is correct! Well done!

You have scored 17 out of 18 for the World Capital's Quiz!
```

Let us say, that prior to playing this game, the high score file looked like this:

```
High Scores for dazh001
1. 18
2. 17
3. 14
4. 13
5. 0
```

After playing the game and scoring 17, the high score file will look like this:

```
High Scores for dazh001
1. 18
2. 17
3. 17
4. 14
5. 13
```