COMPSCI 101 Assignment – Yacht

Due: 11:59pm, Friday 23rd May 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

Submit the file containing your assignment using the Assignment Dropbox:

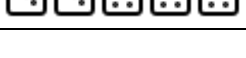
https://adb.auckland.ac.nz/Home/

INTRODUCTION

For this assignment you will be implementing a simplified version of the game of Yacht. Yacht is a dice game where the object of the game is to score points by rolling five dice to make certain combinations called categories. A game consists of six rounds. Each round consists of the player:

- Rolling the five dice.
- Choosing a scoring category.
- Calculating the value of the dice roll based on the scoring category selected and add it to a running total.

Each scoring category can only be used once. There are six categories, so at the end of the game all of them will have been used. The categories are as follows:

Category	Score	Example
Ones	Sum of ones in the roll	 scores 3
Twos	Sum of twos in the roll	 scores 4
Threes	Sum of threes in the roll	 scores 3
Fours	Sum of fours in the roll	 scores 12
Fives	Sum of fives in the roll	 scores 10
Sixes	Sum of sixes in the roll	 scores 18

If a category is chosen but the dice do not match the requirements of the category, the player scores 0 for that category. For example, if the player chooses Ones and rolls  , they will score 0 for the Ones category.

At the end of the game, if the player's score falls within their top five high scores, the list of top five high scores is updated accordingly. The player's top five high scores are stored in a text file.

Your program will need to:

- Display a banner.
- Run the game of Yacht. This will involve six rounds. In each round you need to:
    - Simulate rolling five dice (**the function that does this will be provided for you**).
    - Display the five dice rolled, sorted in ascending order.
    - Get the user to select a scoring category.
    - Calculate the score for the round based on the dice rolled and the category selected.
    - Add the round score to the running total.
- 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 – `Yacht_Skeleton.py`. The file contains:

- The statement `import random`. This statement is required to be able to simulate rolling dice. Do not remove this statement!
- An implementation of the `roll_dice()` function. This function simulates rolling five dice, returning the values rolled as integers in a list. Do not 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()`
    - `calculate_roll_score()`
    - `get_category()`
    - `play_game()`
    - `read_high_scores()`
    - `update_high_scores()`
    - `handle_high_scores()`
    - `main()`

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 `"Yacht by {username}"`. It then prints this message inside a banner. For example, if the function is called passing it the username `"DAZH001"`, then the function would print the following banner:

```
**********************
*  Yacht by DAZH001  *
**********************
```

The function should print a blank line after the banner

`calculate_roll_score()` (4 marks)

The `calculate_roll_score()` function takes two parameters:

1. `category` – an integer representing the scoring category selected by the player. This integer will be a 1, 2, 3, 4, 5, or a 6.
2. `dice_roll` – a list of 5 integer values representing the five dice rolled.

The function will calculate and return the score of the dice roll using the scoring category selected by the player. A summary of how scoring works is given in the **Introduction** section of this handout.

`get_category()` (5 marks)

The `get_category()` function takes a single parameter – a list of integers between 1 and 6 (both inclusive), representing the available scoring categories called `available_categories`. The function will:

- Print the available scoring categories.
- Prompt the player to enter the category they want to choose. You can assume that the player will always enter an integer value.
- While the player enters a category that is not available, the function should inform them of this and prompt them to enter a category again.
- Remove the selected category from the list of available categories.
- Return the selected category.

For example, when the game starts, the list of available categories will be `[1, 2, 3, 4, 5, 6]`.

```
Available categories: [1, 2, 3, 4, 5, 6]
Please choose a category from those available: -1
-1 is not available!
Please choose a category from those available: 7
7 is not available!
Please choose a category from those available: 4
```

In this example the player has chosen the categories -1, 7 and 4. Both -1 and 7 are both invalid categories, so the player is informed of this and prompted to choose a category again. After the player enters a valid scoring category, 4, the function removes this value from the list of available categories and returns it. In the next round of the game, the list of available categories will be `[1, 2, 3, 5, 6]`.

`play_game()` (10 marks)

The `play_game()` function runs the game of Yacht and takes two parameters:

1. `available_categories` – a list of integers between 1 and 6 (both inclusive), representing the available scoring categories.
2. `category_names` – a list of six strings representing the names of the scoring categories. This list will always be `["Ones", "Twos", "Threes", "Fours", "Fives", "Sixes"]`

The `play_game()` function will:

- Keep track of the game's total score. When the game begins the total score will be initialized to 0.
- Run through 6 game rounds. For each round the function:
  - Simulates rolling five dice by calling the `roll_dice()` function. This function has been provided for you.
  - Prints the five dice rolled sorted in ascending order based on their value.
  - Gets the user to choose their scoring category by calling the `get_category()` function.
  - Prints the name of the scoring category the user has chosen.
  - Calculates the score obtained for the round, by calling the `calculate_roll_score()` function.
  - Updates the game's total score by adding to it the score obtained for the round.
  - Prints the score obtained for the round.
  - Prints the current total score if this is not the last round.
  - Prints a blank line.
- Prints a message of congratulations followed by the total score obtained for the game.
- Returns the total score.

```
You have rolled the following: [1, 3, 3, 4, 4]
Available categories: [1, 2, 3, 4, 5, 6]
Please choose a category from those available: 4
You have chosen Fours
You have scored 8 this round.
Your current total score is 8.

You have rolled the following: [1, 1, 1, 2, 3]
Available categories: [1, 2, 3, 5, 6]
Please choose a category from those available: 1
You have chosen Ones
You have scored 3 this round.
Your current total score is 11.

You have rolled the following: [1, 2, 3, 5, 6]
Available categories: [2, 3, 5, 6]
Please choose a category from those available: 2
You have chosen Twos
You have scored 2 this round.
Your current total score is 13.

You have rolled the following: [3, 4, 5, 6, 6]
Available categories: [3, 5, 6]
Please choose a category from those available: 6
You have chosen Sixes
You have scored 12 this round.
Your current total score is 25.

You have rolled the following: [1, 2, 2, 3, 5]
Available categories: [3, 5]
Please choose a category from those available: 3
You have chosen Threes
You have scored 3 this round.
Your current total score is 28.

You have rolled the following: [1, 4, 4, 5, 5]
Available categories: [5]
Please choose a category from those available: 5
You have chosen Fives
You have scored 10 this round.

Congratulations! You have scored 38!
```

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 Yacht 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 below shows an example of what the high score file could look like.

```
High Scores for DAZH001
1. 32
2. 28
3. 21
4. 16
5. 13
```

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 `[32, 28, 21, 26, 13]`.

update_high_scores() (8 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 Yacht game.
2. `username` – A string representing the player's username.
3. `high_scores` – A list of five integer values representing the player's top five high scores for the Yacht game.
4. `new_score` – An integer value representing the player's current score in the Yacht game.

The function should update the list of top five high scores if the player's current score is large 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. 32
2. 28
3. 21
4. 16
5. 13
```

And the player scored 37 in their current game of Yacht, then the updated high score file would look like this:

```
High Scores for DAZH001
1. 37
2. 32
3. 28
4. 21
5. 16
```

If the player's current score is not large 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 Yacht game.
2. `username` – A string representing the word the player's username.
3. `new_score` – An integer value representing the player's current score in the Yacht 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 Yacht program. The first three statements, all of them initialization statements, have been provided for you. Do not alter them! Update the fourth statement so that you initialize the variable `username` to your username (all alphabetical letters should be in **uppercase**).

Complete the rest of the function so that it:

1. Prints the game banner by calling the `print_banner()` function.
2. Runs the game of Yacht and obtains the player's score by calling the `play_game()` function.
3. 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 80 characters.

Marking Rubric

Implementation	40 marks
`print_banner()`	2 marks
`calculate_roll_score()`	4 marks
`get_category()`	5 marks
`play_game()`	10 marks
`read_high_scores()`	6 marks
`update_high_scores ()`	8 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 YACHT GAME BEING PLAYED

```
**********************
*  Yacht by DAZH001  *
**********************

You have rolled the following: [1, 1, 3, 5, 5]
Available categories: [1, 2, 3, 4, 5, 6]
Please choose a category from those available: 0
0 is not available!
Please choose a category from those available: 5
You have chosen Fives
You have scored 10 this round.
Your current total score is 10.

You have rolled the following: [1, 1, 2, 5, 6]
Available categories: [1, 2, 3, 4, 6]
Please choose a category from those available: 1
You have chosen Ones
You have scored 2 this round.
Your current total score is 12.

You have rolled the following: [1, 2, 3, 5, 6]
Available categories: [2, 3, 4, 6]
Please choose a category from those available: 2
You have chosen Twos
You have scored 2 this round.
Your current total score is 14.

You have rolled the following: [1, 2, 3, 4, 6]
Available categories: [3, 4, 6]
Please choose a category from those available: 3
You have chosen Threes
You have scored 3 this round.
Your current total score is 17.

You have rolled the following: [2, 3, 4, 5, 6]
Available categories: [4, 6]
Please choose a category from those available: 4
You have chosen Fours
You have scored 4 this round.
Your current total score is 21.

You have rolled the following: [1, 3, 3, 4, 4]
Available categories: [6]
Please choose a category from those available: 6
You have chosen Sixes
You have scored 0 this round.

Congratulations! You have scored 21!
```

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

```
High Scores for DAZH001
1. 37
2. 32
3. 28
4. 21
5. 16
```

After playing the game and scoring 21, the high score file will look like this:

```
High Scores for DAZH001
1. 37
2. 32
3. 28
4. 21
5. 21
```