Objective 7: Challenges

You should attempt one difficulty 🛠 challenge, one difficulty 🛠🛠 challenge, and one difficulty 🛠🛠🛠 challenge.

Menu selection challenge

Difficulty: 🛠

Write a program that presents a simple menu to the user: 1. Play Game 2. Instructions 3. Quit
The program should ask the user for a number between 1 and 3 and will output "Let's go" only when a valid entry is made, repeating the menu selection until a valid entry is made.

Compound interest challenge

Difficulty: 🛠

Write a program that shows how compound interest grows in a bank savings account. The program should ask the user to input their current balance, the interest rate (0.04 would be 4%) and the desired balance to reach. The program should output the new balance each year until the desired balance is reached. Interest is added to the balance each year.

e.g.

£100 starting balance – 4% interest rate (0.04)
Year 1: New balance = 100 + (100*0.04) = £104
Year 2: New balance = 104 + (104*0.04) = £108.16
Year 3: New balance = 108.16 + (108.16*0.04) = £112.49

Guess the number game challenge

Difficulty: 🛠🛠

The computer guesses a number between 1 and 100. The player has to try and guess the number in as few attempts as possible. When the user enters a number they are told if the guess is too high or too low until the number is guessed correctly. The player is told how many guesses they made. Write a program to play the game.

Gamertag challenge

Difficulty: 🛠🛠

A gamertag is a username on a typical games console that other players can identify the player with. There are some limitations to them, such as, it cannot be longer than 15 characters.

Write a program to input a valid gamertag and keep asking for a new gamertag until a valid entry is made.

Pseudocode
1.0 Write a comment to explain the program checks the length of gamerTags entered.
2.0 Set a variable called valid_gamertag = true
3.0 Start a condition controlled iteration to check valid_gamertag
4.0 Output a message on the screen to say gamertags must be less than 15 characters.
5.0 Get a gamertag from the keyboard and put it into a variable called gamertag.
6.0 Using the function len() put the number of characters in gamertag in a variable called gamertag_length.
7.0 Check if the length of gamertag is more than 15 characters by comparing gamertag_length to the number 15.
7.1 Output a message to say 'your gamertag is too long' if it is of an invalid length.
7.2 Output a message to say 'gamertag accepted' if it is of a valid length.
8.0 End iteration

Rock, paper, scissors challenge

Difficulty: 🛠🛠🛠

The computer and player choose one of rock, paper, or scissors. The output of the encounter is then displayed with rock beating scissors, scissors beating paper, and paper beating rock. The winner scores 1 point for a win. The score for both players should be output. The game is won when one player reaches 10 points.

Happy numbers challenge

Difficulty: 🛠🛠🛠

Write a program to output whether a chosen number is happy or sad. A happy number is a number defined by the following process: Starting with any positive integer, replace the number by the sum of the squares of its digits, and repeat the process until the number either equals 1, or it loops endlessly in a cycle which does not include 1. Those numbers for which this process ends in 1 are happy numbers, while those that do not end in 1 are unhappy numbers (or sad numbers). When the algorithm ends in a cycle of repeating numbers, this cycle always includes the number 4.

For example, 19 is happy, as the associated sequence is:

$1^2 + 9^2 = 82$
$8^2 + 2^2 = 68$
$6^2 + 8^2 = 100$
$1^2 + 0^2 + 0^2 = 1$.

XP Challenge

Difficulty: 🛠🛠🛠

In many online multiplayer games, players are awarded experience points (known as XP) for completing a variety of objectives in the game. When the player has accumulated enough XP, their rank increases (they rank up) and their XP resets to zero (although extra XP is carried over). Each rank requires more XP than the one before.

Write a program that:

- Allows the user to keep entering XP for a game until it is 2000 or more.
- When the XP reaches 100, 'Promoted to Corporal' is output.

Pseudocode
1.0 Set the player's total XP to be 0
2.0 Set the player's rank to be 1
3.0 Loop while the player's XP is less than 2000
3.1 Ask the user to enter the XP earned in the last game
3.2 Add the XP earned from the last game to the total XP
3.3 Check if the XP earned is more than that required to be a Corporal and that their rank is 1
3.3.1 Tell the player they have been promoted
3.3.2 Set XP back to zero but carry over any XP left
3.3.3 Set player's rank to be 2
3.4 Display the total XP

- Extend the program so that the player ranks up according to the table below:

XP needed	**Rank awarded**
0	Private
100	Corporal
300	Sergeant
700	Staff Sergeant
1500	Warrant Officer