

8.11: Worksheet 8

This worksheet has been split into 2 sections: Feasibility and Competitor Analysis, and Programming Practice. As a minimum, you must complete the feasibility and competitor analysis section – there are not many questions as they might take a while to complete. The programming practice section has been added to give you the opportunity to keep building your skills. There is no new content within the programming section, but it does allow you to practice what you have already learnt (this will make it easier when we return to programming).

# 8.11.1: Competitor and Feasibility Analysis

1. Discuss Porter’s Five Forces
2. Investigate how you can carry out a Porter’s Five Forces Analysis
3. What are the benefits and drawbacks of Porter’s Five Forces Analysis
4. Complete a feasibility analysis for a recent innovation in computer science
5. You have been asked to create a new application for a food delivery service. Complete the following activities:
   1. Feasibility Analysis
   2. Porter’s Five Forces Analysis
   3. Competitor Analysis
6. In the slides, there is a set of reflective questions in relation to the debate. Using these questions as prompts, write a short reflection (about 500 words) about the in-class debate.

# 8.11.2: Programming Practice

1. Write a program which:
   1. Asks the user for a list of names
   2. Asks the user for a list of restaurants
   3. Prints a combination of each of these lists to the console, so that each person “eats” at each restaurant. An example output would be:



1. Write a function which tests whether a number is prime. Then use the function to:
   1. Ask the user to enter a number, then tell them whether it is prime or not
   2. Ask the user to enter a number, n, then tell them the nth prime number (for example, if you enter 6 the output should be 13, as the 6th prime number is 13 – 2, 3, 5, 7, 11, 13)
2. Write a function which checks the validity of a password entered by a user. The password must have at least:

* 8 characters
* One upper case letter
* One lower case letter
* One number

1. Create a game where the computer picks a random word and the player must guess the word. The computer tells the player how many letters are in the word. Then the player has a set number of opportunities to guess whether a letter is in the word – with the computer replying yes or no. After each guess, the user should be asked whether they want to guess the word. If the player guesses correctly, the game finishes and there should be a congratulations message together with the number of guesses it took them. If the player fails to guess the word, the computer should commiserate with them and output what the word was.