GETTING STARTED

**Your first C# Program**

\* define a variable called name

\* set the variable to your own name

\* print that variable to the console

**Doing Math**

\* Use two variables, 10 and 3

\* Perform each of the arithmetic operations on them (+ - \* / %)

\* You might be surprised by the quotient (if you’re not getting the expected answer, try declaring them as ‘double’s instead of ‘int’s.

OPERATORS AND STRINGS

**User Input**

\* ask the user for their name (ie: print the question to the console)

\* save the answer to a variable

\* greet the user by name (ie: Hello …)

DECISION MAKING

**The Else If statement**

\* Ask the user who they are (think of 3 possibilities of who the user might be)

\* Ask a question to the user depending on who the user is

\* Print something if you don’t recognise the user (ie: who is not one of those 3 possibilities)

**Logical Operators**

\* You are a bouncer, and will only admit people who -

\* Look over 25, OR

\* Have id AND are over 18.

There are 8 combinations, which you can check by asking the user for and providing for the following information. (This could also be achieved using nested loops, but don’t worry if this seems difficult at this stage)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **HAS ID** | | Looks over 25 | |  | **HAS NO ID** | | Looks over 25 | |
| TRUE | FALSE | TRUE | FALSE |
| Over 18 | TRUE | (1) | (2) | Over 18 | TRUE | (5) | (6) |
| FALSE | (3) | (4) | FALSE | (7) | (8) |

**The switch statement**

\* Do the same as in the “The If Else statement”, but using a switch statement

**The Ternary Operator**

\* Someone is asking where the toilets are. Use the user input to determine their gender.

\* Use the conditional operator to set a Direction variable according to their answer, (left if male and right if female).

\* print the direction they need to go.

LOOPS

**The while loop**

\* Research how to generate a random integer

\* Generate a random integer between 1 and 5

\* Ask the user to guess the number

\* for each wrong guess, prompt the user to try again

\* On a correct guess, congratulate the user.

**Break and Continue (Break)**

\* Create an ATM:

\* Generate a random 4 digit code

\* Give the user 3 attempts to provide the correct code

\* Provide the relevant service if correct within 3 attempts

\* swallow the card if wrong on the third attempt.

**Break and Continue (Continue)**

\* You are admitting children onto a ride with an age restriction

\* iterate through ages 1 – 10 (ie: **for (var i = 1; i <= 10; i++) …** )

\* if heights are less than 7, state that the child is too young, otherwise, continue through to the next child

EXTRA?

Collections (including IEnumerable and List and Arrays) – there’s more on arrays later

Indexing strings and arrays

Dictionaries – this is mentioned later, but could be useful here.

LINQ methods – Select, Where, GroupBy

EXERCISM

You may want to call your method multiple times with different parameters to check all cases.

1. Leap

2. Two fer

3. Space age

4. Grains

5. Perfect Numbers

6. Collatz Conjecture

7. Difference of Squares

8. Armstrong numbers (possible, but hard at this point)

9. Darts (you probably already know that Pythagoras’ theorem will come in useful here)

10. Triangle

11. Raindrops

12. Beer song

13. Sieve