

Computing

Selection

Selection: IF Statement

```
num= 2
If num > 0:
    print(num, "is a positive number")
print("this sentence is always printed")
```

Selection: ELIF Statement

```
score= 92
If num > 90:
    print("expert")
Elif num >70:
    print("advanced")
Else:
    print("novice")
```

Iteration

Iteration: WHILE Loop

```
password = ("1234")
guess = input("Please enter password: ")
while guess != password:
    print("Access denied")
    guess = input("Please try again")
print("Access granted")
```

Iteration: FOR Loop

```
for i in range(0,5):
    print(i)

for x in range (0,5):
    print("Hello World")
    print("nice to meet you")

for i in range(0,15,3):
    print(i)
```

While Loop = repeats uses a condition

For Loop = set number of repetitions

Programming constructs

Sequencing



Creating a set of instructions to complete a task. The order of the instructions is vital to creating a successful algorithm.

Selection



A decision within a computer program when the program decides to move on based on the results of an event e.g. IF, ELIF, ELSE

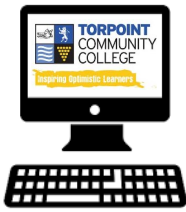
Iteration



Repeating an action. This could be a WHILE loop (do something while a condition is met) or a FOR loop (do something a set number of times).

Logic Error: this will happen when your code does what you told it to do, but not what you wanted it to do, like adding two numbers instead of subtracting them. Logic errors might not crash your program but will cause it to do the wrong thing.

Syntax Error: happen when you break the rules of the language like accidentally missing out a " or). Syntax errors stop your code from running at all.



Computing

Arithmetic operators

Arithmetic	Operator	Example
Addition	+	2 + 10
Subtraction	-	9 - 6
Multiplication	*	5 * 4
Division	/	5 / 2
Floor division	//	7 // 2
Remainder	%	7 % 3

Random Numbers

```
import random
print(random.randint(1, 10))
```

Boolean Operators

AND

Returns True if both statements are true

OR

Returns True if one of the statements is true

NOT

Reverse the result, returns False if the result is true

Variables and Constants

Variable— a values which may change while the program is running.

Local variable— a variable which can only be used within the structure they are declared in

Global Variable— a variable which can be used in any part of the code after they are declared.

Constant— a value which cannot be altered when the program is running.

Comparison operators

Operator	Meaning	Example
==	equal to	7==7
!=	not equal to	6!=7
>	Greater than	7>6
<	Less than	5<8
>=	Greater than or equal to	6>=8
<=	Less than or equal to	7<=7

Data Types

String	Text e.g. "Hello"
Integer	Whole number e.g. 39
Float/Real	Decimal number e.g. 13.5
Boolean	Two values e.g true or false
Character	A single character e.g B
Casting	Is when you want to change between data types

Subroutines

Sometimes called sub programs, they are sets of instructions stored under one name. Subroutines can be functions or procedures— the main difference is that function always return a value and procedures do not.

Subroutines are useful when they have sets of instructions you need to repeat in different places in the program. They give the program more structure and readability whilst cutting down on the amount of code you actually need to write.