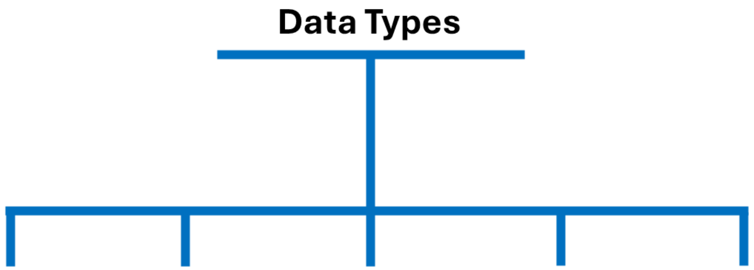
Programming Basics

Do Now Activity (DNA):
Have your answers ready!



What are the 5 main data types we use in programming?

Date:

Summary

Pseudocode

- Pseudocode is used to write instructions in statements that are somewhere between English and a programming language
- There are guidelines for writing pseudocode, but no strict rules
- They are an aid to thinking out the steps needed before you start to code
- Once you have written the pseudocode, the coding should be a breeze!

Pseudocode statements

- Assignment statements are written using an = sign
- The symbols +, -, *, / , ** (exponentiation) are used for common arithmetic operations

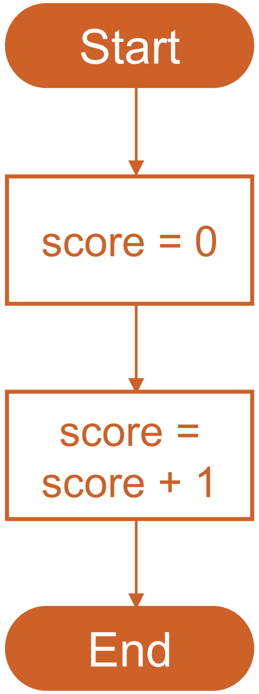
```
miles = currentMileage – previousMileage
count = count + 1
```

- Input and output statement are written

```
currentMileage = input (“Enter current mileage”)
print (“You have travelled ”, miles, “miles”)
```

Variables and assignment

- An **identifier** is a name, e.g. **score**, that points to a memory location
- **Assignment** is assigning a value to the memory location
- In this example **score** is assigned the value of **0**
- In the next line, **1** is added to **score** and the result is again assigned to **score**



Start

score = 0

score = score + 1

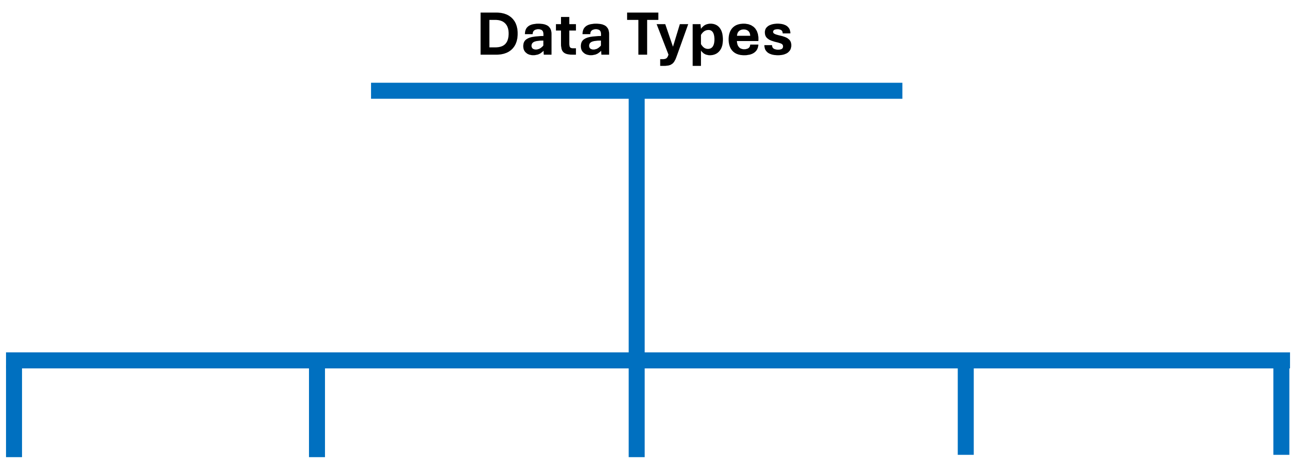
End

Variables and Constants

- The value of a variable can be changed while the program is running
- To change the value of a constant, you have to change it in the source code and then recompile
- A constant cannot be the target of an assignment

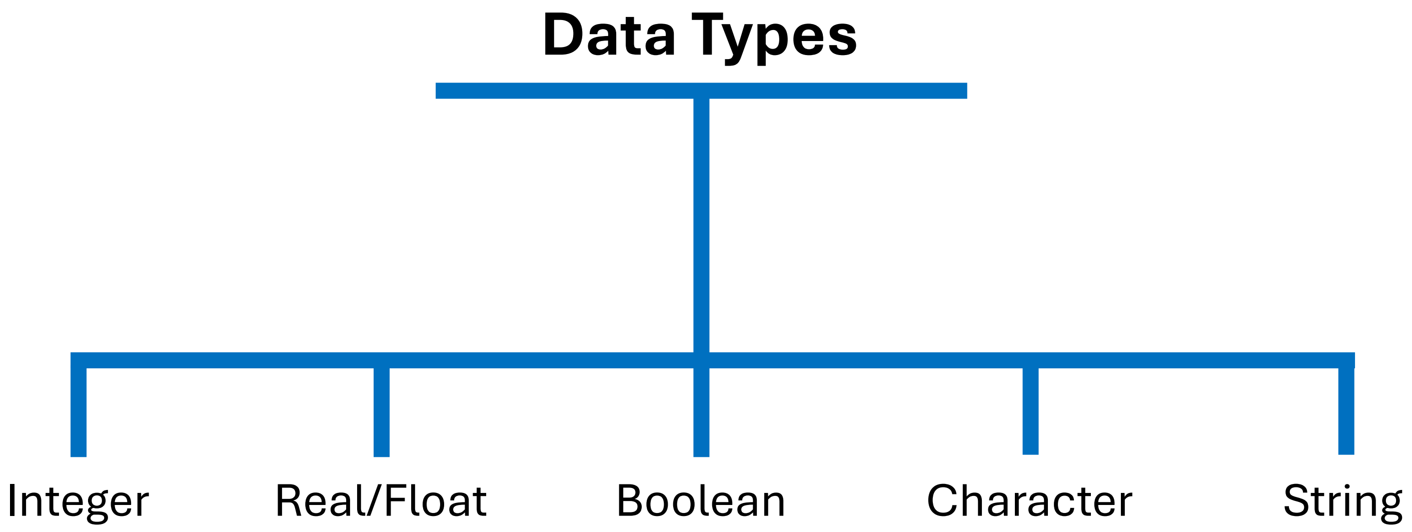
```
Dim name as String = “Robert”
Const VAT as Single = 17.5

Console.WriteLine(“VAT is: “ & VAT)
name = “Hazel”
Console.WriteLine(“Name is: “ & name)

VAT = 20            this will cause an error
```

Data Types



What are the 5 main data types we use in programming?

Data Types



Data Types

Integer

Real/Float

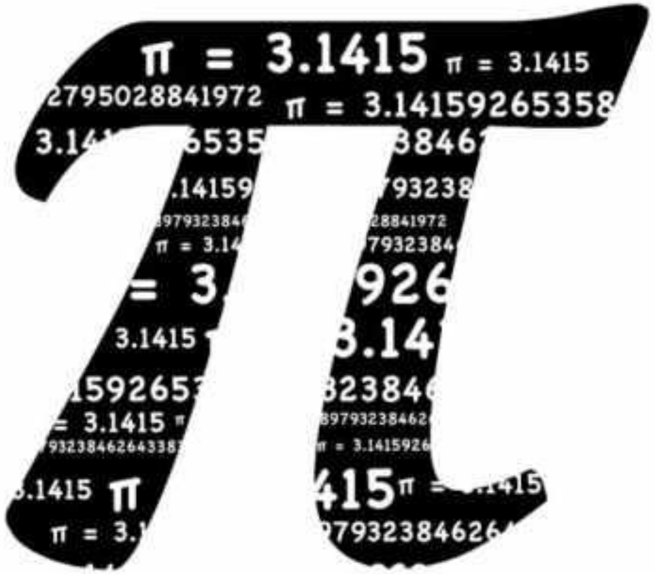
Boolean

Character

String

Why use a constant?

- Constants reduce the risk of errors by reducing access to the memory location
  - Would you declare 3.14159265359 as a variable or a constant?



Programming Basics

Task 3

3. Write pseudocode for a program which calculates the number of miles per gallon a car is doing. The user will input

   - the car mileage the last time the car was filled
   - the car mileage now
   - the total number of litres taken to fill the tank

   n.b. There are 0.22 gallons in a litre, or 4.546 litres in a gallon

   Which of the identifiers in your program could you define as

   (i) a constant? (What is the advantage of doing this?)

   (ii) an integer

   (iii) a real (decimal) number?

The mod and div operators

- The operator mod is used to find the remainder in integer division
- div returns the integer part of the division

```
x = 17 mod 3      sets x = 2
y = 17 div 3      sets y = 5
```

The Round function

- You can round this number using a function **round.**

```
billBetween3 = round(billBetween3,2)   // round to 2 decimal places
```

String handling functions

- Programming languages have a number of built-in string handling functions or methods
- For example:

```
name = "Robert "
```

`x = len(name)` *returns the length of string name*

`x = name.find("be")` *determines if "be" occurs in string name (returns the position of the first character in the string, 0 if not found)*

`ord(“a”)` *returns the integer value of a character (97 in this example)*

`chr(97)` *returns the character represented by an integer (“a” in this example)*

String conversion operations

`int(“1”)` *converts the character “1” to the integer 1*

`str(123)` *converts the integer 123 into a string “123”*

`float(“123.456”)` *converts the string “123.456” to the real number 123.456*

`str(123.456)` *converts the real number 123.456 to the string “123.456”*

`date(year,month,day)` *returns a number that you can calculate with*

Programming Basics

We use a variety of different platforms at RGS for A Level Computer Science. These include:

- Time2Code (transition from GCSE to A Level)
- Trinket (for in lesson programming activities)
- Codio (for in lesson programming activities and preparing for NEA)

Python

Course

All the fundamentals of programming in Python for GCSE examinations.

Programming Basics

Due to restrictions on external applicants having access to the network and some platforms, we are going to complete our practical programming activities on Time2Code for today.

This will also set you up nicely for what we will be doing as soon as we start in September, and to set you on the path nicely for the summer work….(more to come on this at the end of the lesson!)

Python

Course

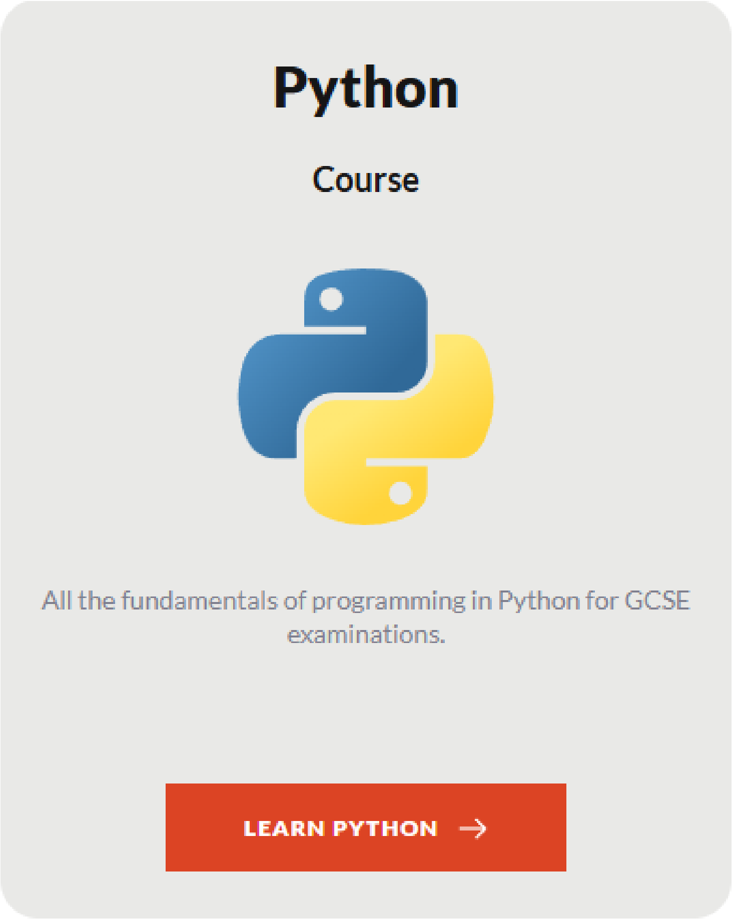
All the fundamentals of programming in Python for GCSE examinations.

Programming Basics

Please now visit www.time2code.today

Please click on the ‘Learn Python’ Module

As you have will need to be able to program to at least a basic GCSE standard, please start by completing ‘Assessment point Levels 1-3’. This will act as a recall of what you have done with programming to this point

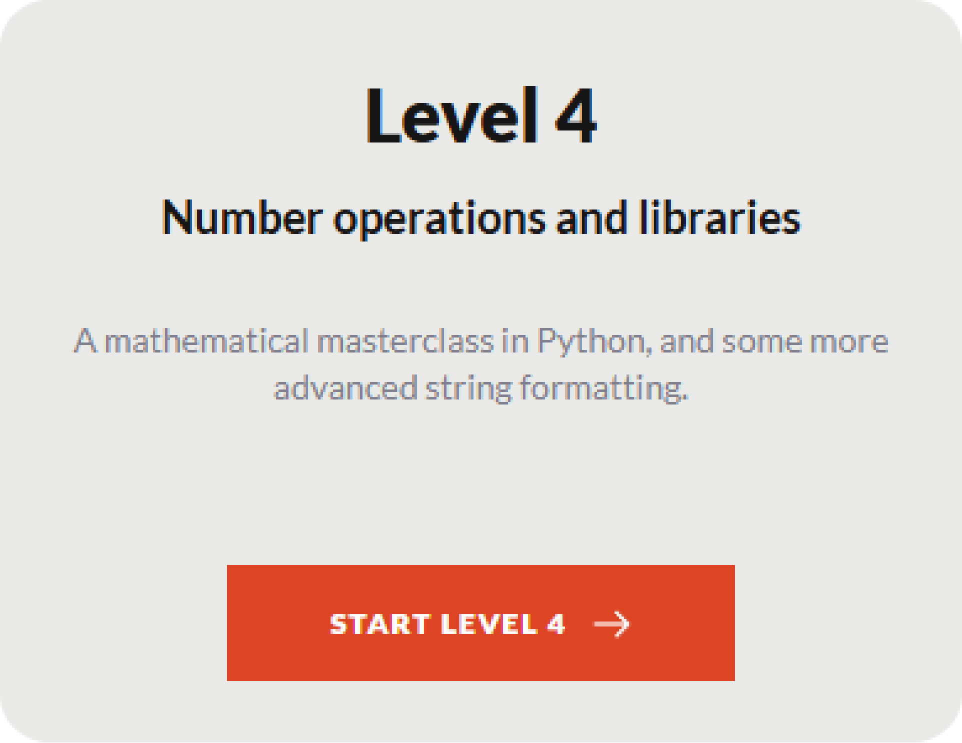


Python

Course

All the fundamentals of programming in Python for GCSE examinations.

Programming Basics



Level 4

Number operations and libraries

A mathematical masterclass in Python, and some more advanced string formatting.

Now we will get you started on some of the more A Level standard programming – using libraries!

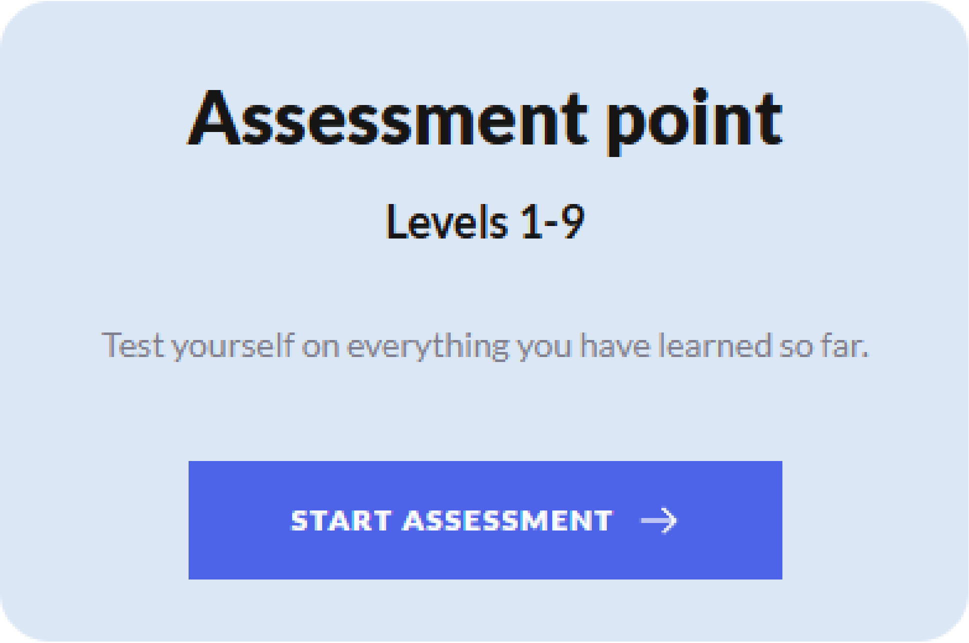
Open Level 4 – Polyhedral Dice

Complete the ‘Try’ and ‘Investigate’ tasks before then having a go at programming a solution to the activity under the ‘Make’ section.

If you finish Level 4 – please continue on your journey today in Time2Code!

Programming Basics

SUMMER WORK!



Assessment point

Levels 1-9

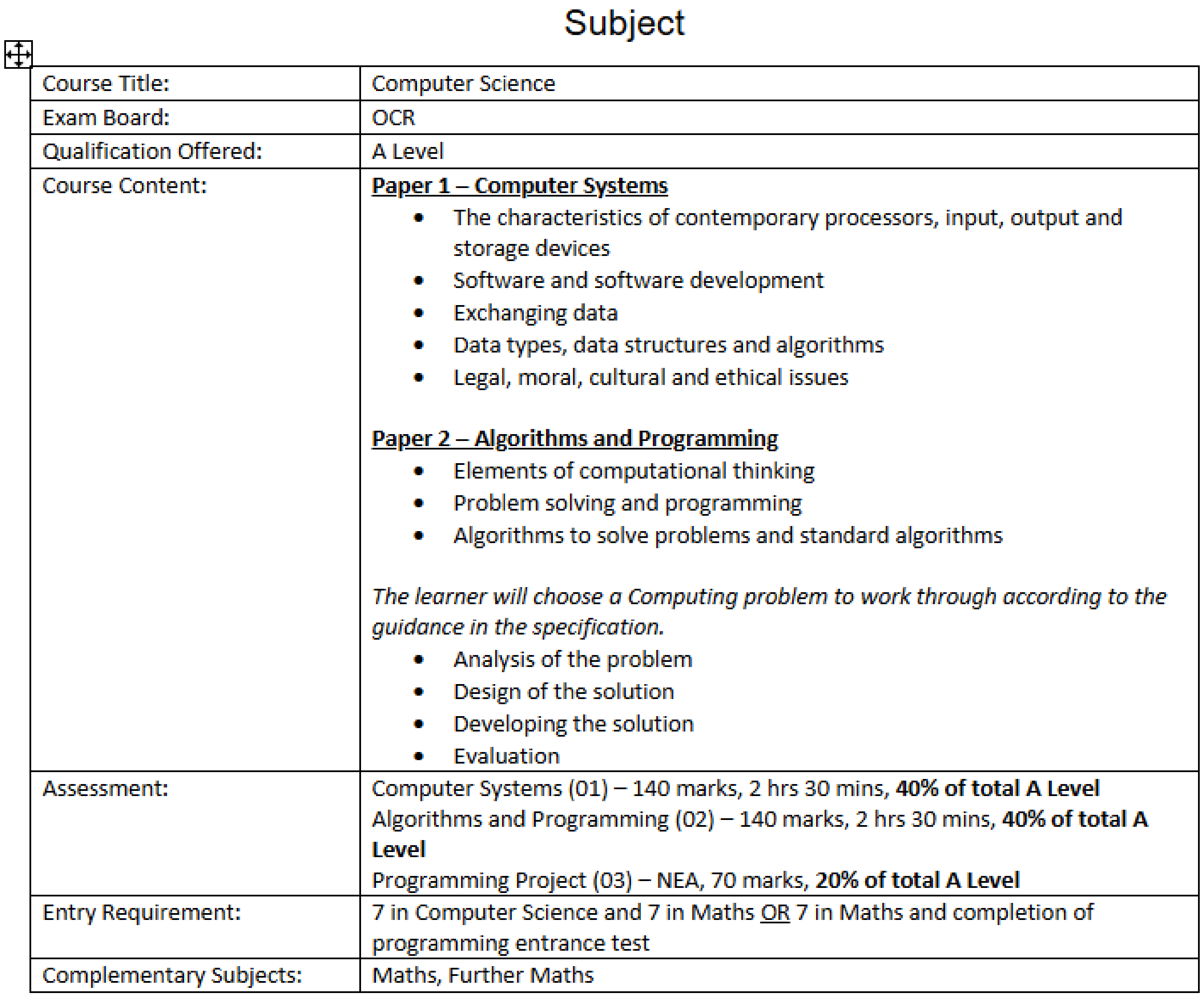
Test yourself on everything you have learned so far.

<u>Time2Code – Levels 1-9 Assessment Point</u>

We are asking our prospective Yr12 students to have completed the activities up until, and including, the end of Assessment Point Levels 1-9 before the first lesson back in September.

This is so that you are fully prepared for the first unit of work in September, which you have had a flavour of today – Programming Techniques.

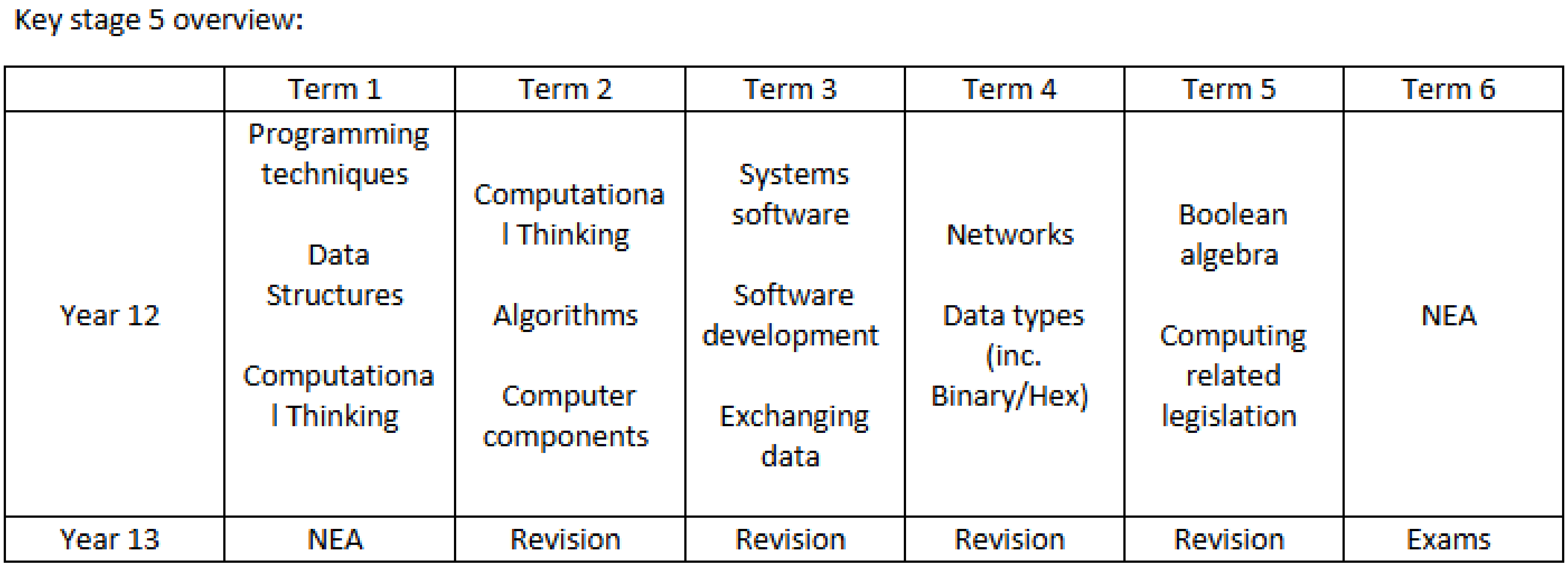
Programming Basics



Subject

Course Title:	Computer Science
Exam Board:	OCR
Qualification Offered:	A Level
Course Content:	**<u>Paper 1 – Computer Systems</u>** • The characteristics of contemporary processors, input, output and storage devices • Software and software development • Exchanging data • Data types, data structures and algorithms • Legal, moral, cultural and ethical issues **<u>Paper 2 – Algorithms and Programming</u>** • Elements of computational thinking • Problem solving and programming • Algorithms to solve problems and standard algorithms *The learner will choose a Computing problem to work through according to the guidance in the specification.* • Analysis of the problem • Design of the solution • Developing the solution • Evaluation
Assessment:	Computer Systems (01) – 140 marks, 2 hrs 30 mins, **40% of total A Level** Algorithms and Programming (02) – 140 marks, 2 hrs 30 mins, **40% of total A Level** Programming Project (03) – NEA, 70 marks, **20% of total A Level**
Entry Requirement:	7 in Computer Science and 7 in Maths <u>OR</u> 7 in Maths and completion of programming entrance test
Complementary Subjects:	Maths, Further Maths

Programming Basics



Key stage 5 overview:

	Term 1	Term 2	Term 3	Term 4	Term 5	Term 6
Year 12	Programming techniques Data Structures Computational Thinking	Computational Thinking Algorithms Computer components	Systems software Software development Exchanging data	Networks Data types (inc. Binary/Hex)	Boolean algebra Computing related legislation	NEA
Year 13	NEA	Revision	Revision	Revision	Revision	Exams