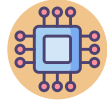


 Menu**TEACH COMPUTER SCIENCE** Menu

Programming Data Types

GCSE Computer Science Resources

14-16 Years Old

48 modules covering every Computer Science topic needed for GCSE level, and each module contains:

- An editable PowerPoint lesson presentation
- Editable revision handouts
- A glossary which covers the key terminologies of the module
- Topic mindmaps for visualising the key concepts
- Printable flashcards to help students engage active recall and confidence-based repetition
- A quiz with accompanying answer key to test knowledge and understanding of the module

[View the GCSE Resources →](#)

KS3 Computing Resources

11-14 Years Old

We've created 45 modules covering every Computer Science topic needed for KS3 level, and each module contains:

- An editable PowerPoint lesson presentation
- Editable revision handouts
- A glossary which covers the key terminologies of the module
- Topic mindmaps for visualising the key concepts
- Printable flashcards to help students engage active recall and confidence-based repetition
- A quiz with accompanying answer key to test knowledge and understanding of the module

[View the KS3 Resources →](#)

[Home](#) / [Programming](#) / Programming Data Types

A **data type** is a means of classifying the type of data that a variable or object can hold in computer programming. Data types are an important factor in all computer programming languages, including C#, C++, JavaScript, and Visual Basic. When programmers develop computer software—whether that's desktop or web-based—data types must be assigned and applied correctly in order to guarantee proper outcomes and an error-free program.

In programming, a data type is a categorization that specifies what kind of operation can be applied to it without resulting in an error.

The data type determines which operations can be safely executed to develop, transpose, and apply the variable to another computation. When a programming language needs a **variable** to be used only in ways that follow its data type, that language is said to be *strongly typed*. Doing this reduces errors, because while it is reasonable to ask the computer to multiply a float by an integer (3.4 x 3), it is unreasonable to ask the computer to multiply a float by a string (3.4 x Sam). When a programming language allows a variable to assume a different data type, the language is said to be *weakly typed*.

Strongly typed and weakly typed programming languages are commonly-held misconceptions. There is a concept of type safety, whereby a programming language constrains or restricts type error.

Common Data Types

- **Integer** – a whole number that can have a positive, negative, or zero value. It cannot be a fraction, nor can it include decimal places. It is commonly used in programming, especially for increasing values. Addition, subtraction, and multiplication of two integers results in an integer. However, division of two integers may result in either an integer or a decimal. The resulting decimal can be rounded off or truncated in order to produce an integer.
- **Character** – any number, letter, space, or symbol that can be entered in a computer. Every character occupies one byte of space.
- **String** – used to represent text. It is composed of a set of characters that can include spaces and numbers. Strings are enclosed in quotation marks to identify the data as strings, and not as variable names, nor as numbers.
- **Floating Point Number** – a **number** that contains decimals. Numbers that contain fractions are also considered floating-point numbers.
- **Array** – a kind of a list that contains a group of elements which can be of the same data type as an integer or string. It is used to organise data for easier sorting and searching of related sets of values.
- **Varchar** – as the name implies, a varchar is a variable character, on account of the fact that the memory storage has variable length. Each character occupies one byte of space, plus 2 bytes additional for length information.

Note: Use Character for data entries with fixed lengths, like phone numbers, but use Varchar for data entries with variable lengths, like address.

- **Boolean** – used for creating true or false statements. To compare values the following operators are being used: AND, OR, XOR, and NOT.

Boolean Operator	Result	Condition
x AND y	True	If both x and y are True
x AND y	False	If either x or y is False
x OR y	True	If either x or y, or both x and y are True
x OR y	False	If both x and y are False
x XOR y	True	If only x or y is True
x XOR y	False	If x and y are both True or both False
NOT x	True	If x is False
NOT x	False	If x is True

- **Date, Time and Timestamp** – these data types are used to work with data containing dates and times.

Date Type	Value
DATE	Year, Month and Day
TIME	Hour, Minute and Second
TIMESTAMP	Year, Month, Day, Hour, Minute, Second and Microsecond

- < Simplex, Half Duplex, Full Duplex
- > Synchronous and Asynchronous Data Transmission

About this site

Teach Computer Science provides detailed and [comprehensive teaching resources](#) for the new 9-1 GCSE specification.

Over 5,000 teachers have signed up to use our materials in their classroom.

What do we provide?

In short: **everything you need to teach GCSE Computer Science:**

- Condensed revision notes for each board
- Exam question booklets

- [Mind maps](#)
- [Interactive quizzes](#)
- [PowerPoint presentations](#)
- [Flashcards](#)

Exam boards

Our materials cover both UK and international exam board specifications:

- [AQA](#)
- [CIE](#)
- [Edexcel](#)
- [OCR](#)
- [Eduqas](#)
- [WJEC](#)
- [National 5](#)

© 2020 Teach Computer Science

This site uses cookies to improve your experience. To find out more, see our [cookie policy](#).