Programming - AQA Computer Science Cheat Sheet

by [deleted] via cheatography.com/56036/cs/15020/

Structured Programming

The **structured approach to programming** means that code is more logical and readable, making it easier to debug and maintain. It also makes it more efficient. The most important part of this approach is **modularised programming**, which is the use of subroutines. Other components include detailed **code annotation** and clearly named variables.

Data Types

Data Type	Definition	Example(s)
Integer	A whole, positive number.	134
Real or **Float**	Any number, including decimals and negatives.	-19.21
Character	A letter, number, space, etc. that can be typed, in accordance with ASCII or Unicode.	@
String	A string of characters.	"Hello, world!"
Boolean	Logical values based on binary (1 and 0).	True, False

Data Type: A specific type of value (and variable) that must be handled in a certain way.

Relational Operators

Symbol	Meaning
=	equal to
≠	not equal to
<	less than
>	greater than
≤	less than or equal to
≥	greater than or equal to

Maths (in Python)

Operation	Example
Addition	**Pseudocode**: 11 + 2 = 13 **Python example**: `11 + 2 = 13`
Subtraction	**Pseudocode**: 11 - 2 = 9 **Python example**: `11 - 2 = 9`
Multiplication	**Pseudocode**: 11 x 2 = 22 **Python example**: `11 * 2 = 22`
Real or Float Division	**Pseudocode**: 11 DIV 2 = 5.5 **Python example**: `11 / 2 = 5.5`
Integer Division	**Pseudocode**: 11 // 2 = 5 **Python example**: `11 // 2 = 5`
Remainder of a Division	**Pseudocode**: 11 MOD 2 = 1 **Python example**: `11 % 2 = 1`

Selection

Selection is when a program makes a decision, usually taking the form of IF statement.
For example (pseudocode):

```
IF x = TRUE THEN
  PRINT " Yes."
ELSE
  PRINT " No."
```

Variables

What is the 'scope' of a variable?

> The part of the program where a variable is valid and accessible.

Why is it important to give variables distinctive identifiers?

> This makes debugging and maintenance easier, as it is easier for the programmer to understand the purpose of the variable.

Programming - AQA Computer Science Cheat Sheet

by [deleted] via cheatography.com/56036/cs/15020/

Robust & Secure Programming

There are two main ways to make programs more robust.
Firstly, using **data validation**, which is most easily done using an IF/ELIF/ELSE statement or a TRY/EXCEPT statement. This makes sure the data inputted by the user is valid, preventing a runtime error.
Secondly, **authentication**. This primarily takes the form of passwords.
However, **testing** is also key in ensuring that a program is robust. A programmer must test typical (normal), boundary (extreme), and erroneous data to guarantee that the program deals with data correctly.

Data Structures

What are **data structures**?

A data structure is a way of storing data.

What is an **array**?

An array is a collection of related data (of the same data type). Each piece of data is an **element** with a specific **index**. They may also be called 'lists'.

What is a **two dimensional array**?

Essentially, a 2D array is just an array made of arrays. They can be thought of as a matrix. Possible uses include a bitmap for an image.

What is a **record**?

A record is very similar to an array, except multiple data types can be stored together.

Programming Languages

High-Level Languages vs Low-Level Languages

Most programs are initially written in high-level programming languages (e.g. Python) because these are more like human languages, meaning that they are easier for programmers to understand, code in, and debug. They are said to provide a 'higher level of abstraction' from machine code. Low-level langages are very different and much more difficult. Despite this, programming in machine code allows for the optimisation of code and avoid code having to be translated.

Programming Languages (cont)

Machine Code vs Assembly Language

Both machine code and assembly language are low-level languages, with assembly code having a 1:1 correspondance with machine code. However, each type of processor has its own specific machine code instruction set, which is expressed in binary. Assembly language is generally used for software for embedded systems and for controlling specific hardware components.

Translation

All programs written in high-level languages or assembly language must be translated into machine code before they can be run.

Interpretors, compilers, and assemblers

An **assembler** translates assembly language to machine code. An **interpretator** translates high-level languages into machine code. It does so **line-by-line**, which makes debugging easier. A **compiler** also translates high-level languages to machine code, but it **translates the whle program before running**. While the compiliation process is slow, the machine code can then be stored and run quickly in future.

Types of Iteration

What is **definite iteration**?

There is a set number of iterations.

Example of definite iteration (Python):

```
for i in range(5):
   x = x + 1
```

What is **indefinite iteration**?

There is not set number of iterations, as it is instead dependent on when a certain condition becomes true.

Example of indefinite iteration (Python):

```
while x < 5:
   x = x + 1
```

What is **nested iteration**?

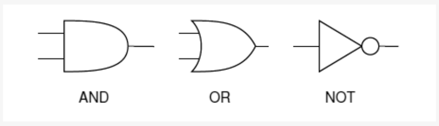
A loop (iteration) within a loop.

Examples of nested iteration (Python):

```
while n == False:
   for i in range(5):
     x = x + 1
```

Note that selection statements can also be nested.

Boolean Logic Gates



SOURCE: www.cs.hmc.edu/csforall/ComputerOrganization/ComputerOrganization.html

Subroutines

What is are **subroutines**?

Named 'out of line' blocks of code that are executed (called) by writing its name in a program statement.

Advantages of subroutines

1. It makes programs **more efficient**, because blocks of code can be reused.
2. It means that code is **easier to debug**, because it is shorter.

Parameters

A value that is passed to a subroutine. These may also be called 'arguments'.

Procedures vs **Functions**

Both take parameters, but functions return a value, whereas procedures don't.

Local Variables

Subroutines can declare local variables, the scope of which is limited to the subroutine. Using local variables is good practice because it allows the program to be simpler. **Global variables**, on the other hand, make a program much more complex, as they may change frequently.

String Handling

Pseudocode	Python Example	Purpose
LEN()		Find the length of a string.
POSITION()	print(string_to_search.find("a"))	To find the position (similar to an index) of a specific character in a string. If the character appears multiple times, only the first instance will be returned.
SUBSTRING(IntExp, IntExp, StringExp)		To create a substring. The first paramter indicates the start of the substring, the second the end, and the final parameter being the string itself.
+	firstname + lastname	To concatenate (join) two strings together.
ORD()	ASCIIcode = ord('a')	To convert a character to character code.
CHR()	character = chr(97)	To convert character code to character.
STR()	str(birthdate)	To convert a string to an integer.
INT()	int(phonenumber)	To convert an integer to a string.

Note that the pseudocode is not set. Different people will write their psuedocode differently.

Programming - AQA Computer Science Cheat Sheet

by [deleted] via cheatography.com/56036/cs/15020/

Input, Output, and File Handling (Python)

In Python, the input() function allows a program to receive data from the user. Similarly, print() allows the program to output information to the user.
File handling is slightly more complex. A typical file-opening statement would take this format:

```
file_o bject  = open(“ fil ename”, “mode”)
```

The 'mode' could be 'read' ('r' - allows access only), 'write' ('w' - allows access and editing), or 'appending' ('a' - allows information to be added to the end of the file.
Once the program has finished using a file, the "filename".close() function should be used, as this frees up resources.

Random Number Generation

In Python, random number generation requires the importation of the RANDOM module. For example:

```
import random
x = random.ra ndi nt( 0,10)
print(x)
```