OCR GCSE Computer Science

Your notes

Programming Fundamentals & Data Types

Contents

Fundamental Programming Concepts


Your notes

Taking an algorithm and turning it into code, in any language, requires an understanding of several basic programming concepts such as:

- **Variables**
- **Constants**
- **Assignment**
- **Operators**
- **Inputs**
- **Outputs**

Variables, Constants & Assignments

What is a variable?

- A variable is a **named memory location** that **holds data** that during the execution of a program, the **data can change**
- Variables can store a variety of **different types of data** such as numbers, text or true/false values
- To store data in a variable, the process of **assignment** is used

What is a constant?

- A constant is **fixed data** that during the execution of a program **cannot change**
- A constant can store a variety of different types of data, **similar to variables**
- **Pi** is an example of a mathematical **fixed value** that would typically be stored as a **constant**

What is assignment?

- Assignment is the process of **storing data in a variable or constant** under a **descriptive name**
- Assignment is performed using the '**=**' symbol

Assigning variables & constants

Concept	OCR exam reference	Python

Variables	x = 3 name = "Save My Exams"	x = 3 name = "Save My Exams"
Constants	const vat = 0.2 const pi = 3.142	VAT = 0.2 PI = 3.142


Your notes

Operators, Inputs & Outputs

What is an operator?

- An operator is **a symbol used to instruct a computer to perform a specific operation** on one or more values
- Examples of common operators include:
  - **Arithmetic**
  - **Comparison**
  - **Boolean (AND, OR and NOT)**

Arithmetic

Operator	OCR exam reference	Python
Addition	+	+
Subtraction	-	-
Multiplication	*	*
Division	/	/
Modulus (remainder after division)	MOD	%
Quotient (whole number division)	DIV	//
Exponentiation (to the power of)	^	**

Comparison

Operator	OCR exam reference	Python
Equal to	==	==
Not equal to	!=	!=
Less than	<	<
Less than or equal to	<=	<=
Greater than	>	>
Greater than or equal to	>=	>=

Examples

Operator	OCR exam reference	Python
Addition	sum = 2 + 2 # 4	sum = 2 + 2 # 4
Multiplication	sum = 3 * 4 # 12	sum = 3 * 4 # 12
Modulus	sum = 10 MOD 3 # 1	sum = 10 % 3 # 1
Quotient	sum = 10 DIV 3 # 3	sum = 10 // 3 # 3
Exponentiation	sum = 2 ^ 2 # 4	sum = 2 ** 2 # 4
Equal to	if 3 == 3 then # True	if 3 == 3: # True
Not equal to	if 5 != 6 then # True	if 5 != 6: # True
Greater than or equal to	if 10 >= 2 then # True	if 10 >= 2: # True

Your notes

AND	number = 10 if number > 0 and < 20 then # True	number = 10 if number > 0 and number < 20: # True


Your notes

What is an input?

- An input is a value that is **read from an input device** and then **processed** by a computer program
- Typical input devices include:
  - **Keyboards** - Typing text
  - **Mice** - Selecting item, clicking buttons
  - **Sensors** - Reading data from sensors such as temperature, pressure or motion
  - **Microphone** - Capturing audio, speech recognition
- Without inputs, programs are **not useful** as they can't i**nteract with the outside world** and always produce the **same result**

What is an output?

- An output is a value **sent to an output device** from a computer program
- Typical output devices include:
  - **Monitor** - Displaying text, images or graphics
  - **Speaker** - Playing audio
  - **Printer** - Creating physical copies of documents or images

Area of a rectangle program

OCR exam reference

```
# Get the length and width from the user
length = input("Enter the length of the rectangle: ")
width = input("Enter the width of the rectangle: ")

# Calculate the area
area = length * width

# Check if the area is greater than 100
if area > 100 then
   # Print the results
```

```
   print("The area of the rectangle is", area)
endif
```

Python

```
# Get the length and width from the user
length = int(input("Enter the length of the rectangle: "))
width = int(input("Enter the width of the rectangle: "))

# Calculate the area
area = length * width

# Check if the area is greater than 100
if area > 100:
   # Print the results
   print("The area of the rectangle is", area)
```

Your notes

Worked Example

A cinema calculates ticket prices based on age category

- Adult = £13.00
- Child = £7.50

The program asks the user to enter their age and calculates the cost of their ticket

A simple algorithm is used

```
adult = 13.00
child = 7.50
age = input("What is your age: ")
if age > 18 then
  total_cost = adult
else
  total_cost = child
end if
print(total_cost)
```

The cinema decides to add a discount of 25% to customers who come to the cinema on 'Sunday evening'

Identify **all** the additional inputs that will be required for this change to the algorithm **[2]**

How to answer this question

- What new information is needed?

Answer

- **day**
- **time**

Your notes

Programming Constructs

Your notes

What is a Programming Construct?

- A programming construct determines the **order in which lines of code are executed**
- They **control logic and behaviour** of code
- There are three core programming constructs:
  - **Sequence**
  - **Selection**
  - **Iteration**

Sequence

What is sequence?

- Sequence refers to lines of code which are run **one line at a time**
- The lines of code are run in the **order** that they written from the first line of code to the last line of code
- Sequence is **crucial to the flow** of a program, any instructions out of sequence can lead to **unexpected behaviour or errors**

Example

- A simple program to ask a user to input two numbers, number two is subtracted from number one and the result is outputted

Line	OCR reference language/Python
01	print("Enter the first number")
02	Num1 = input()
03	print("Enter the second number")
04	Num2 = input()
05	Result = Num1 - Num2

06	print(Result)

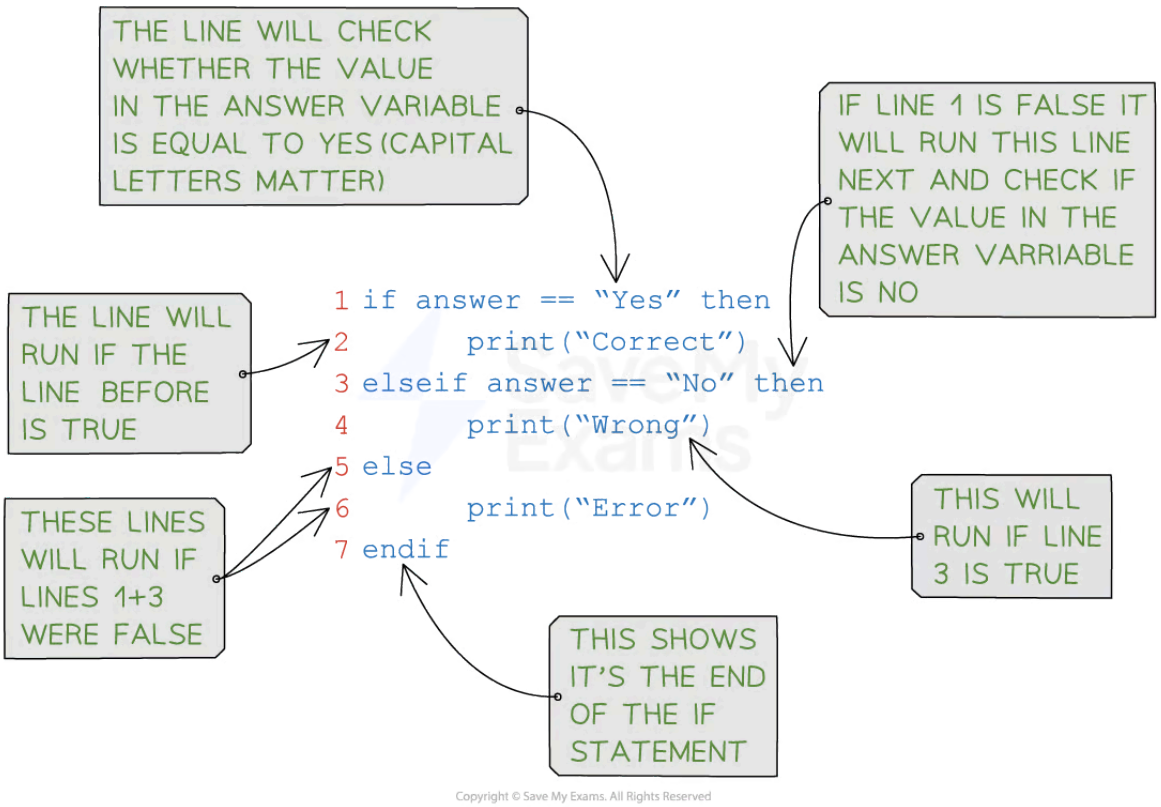
Your notes

- A simple swap of line 01 and line 02 would lead to an unexpected behaviour, the user would be prompted to input information without knowing what they should enter

Selection

What is selection?

- Selection is when the **flow of a program is changed,** depending on a **set of conditions**
- The outcome of this condition will then determine **which lines or block of code is run next**
- Selection is used for **validation**, **calculation** and making sense of a **user's choices**
- There are two ways to write selection statements:
  - **if**... **then**... **else**... statements - this is when you test conditions sequentially
  - **case select** or **switch**... statements - this is when you test an expression against multiple possible constant values (known as cases)



Example

Concept	OCR exam reference	Python

IF-THEN-ELSE	if answer == "Yes" then
	print("Correct")
	elseif answer == "No" then
	print("Wrong")
	else
	print("Error")
	endif
		if answer == "Yes":
		print("Correct")
		elif answer == "No":
		print("Wrong")
		else:
		print("Error")

CASE SELECT or SWITCH	switch day : case "Sat": print("Saturday") case "Sun": print("Sunday") default: print("Weekday") endswitch	match day: case "Sat": print("Saturday") case "Sun": print("Sunday") case _: print("Weekday")

Your notes

If vs select case

- **Select** case **can mean less code** but it only useful when comparing **multiple values of the same variable**
- **If** statements can be **more flexible** and are generally used more in languages such as Python

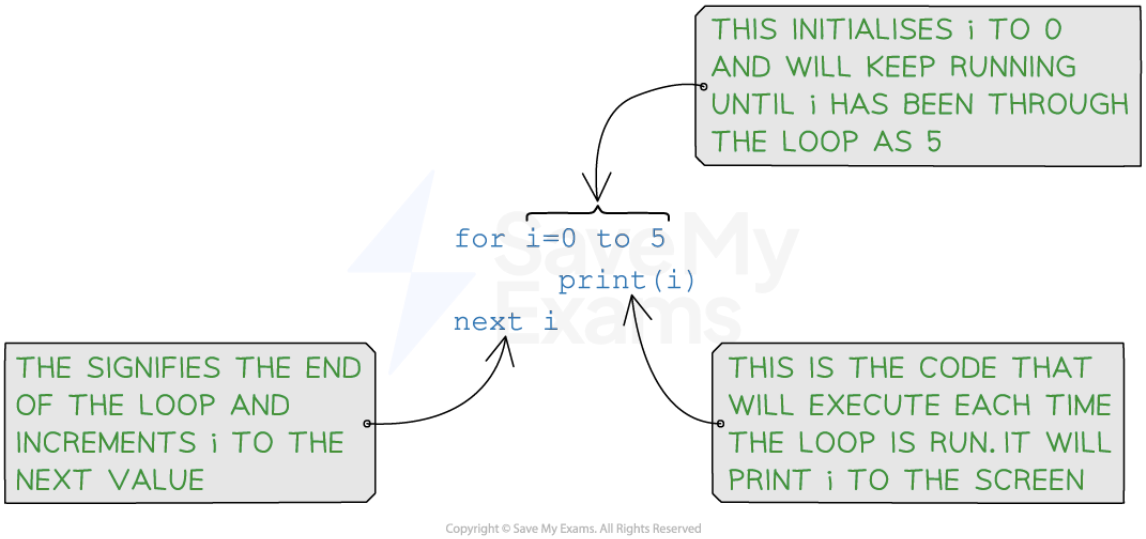
Iteration

What is iteration?

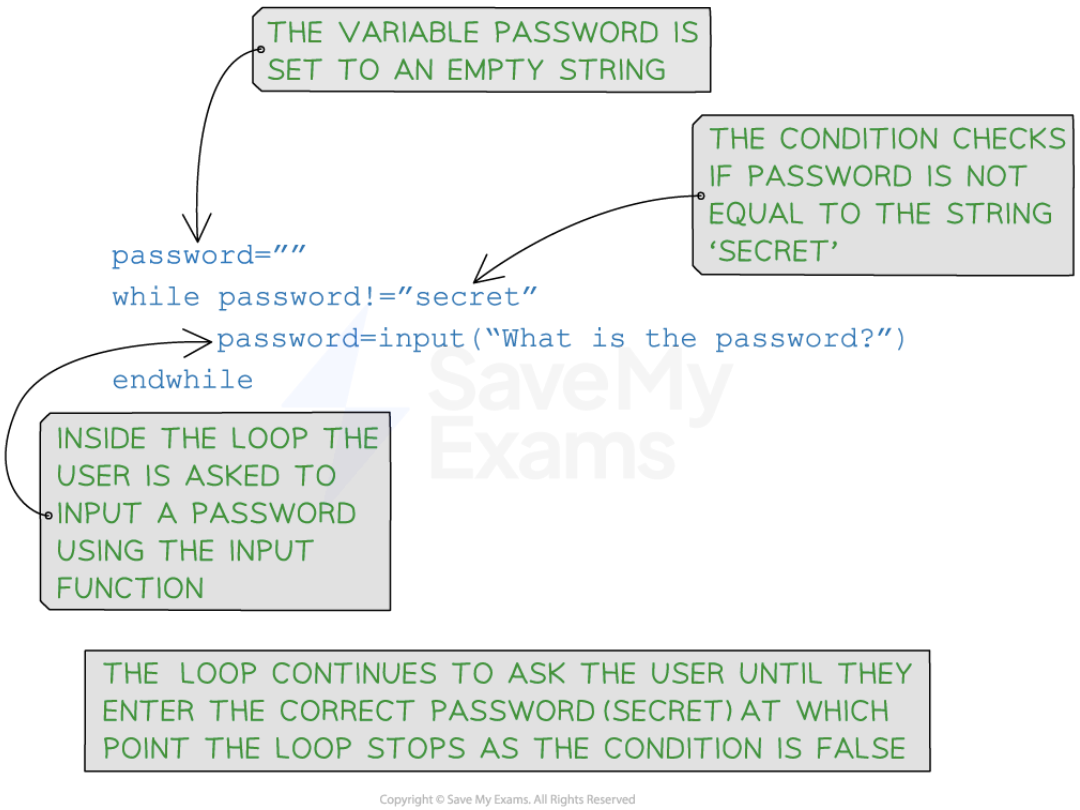
- Iteration is **repeating a line or a block of code** using a loop
- Iteration can be:

Your notes

- **count controlled** - this is when the code is repeated a fixed number of times (e.g. using a for loop)



- **condition controlled** - this is when the code is repeated until a condition is met (e.g. using a while loop or a do while loop)



Your notes

Examples

Iteration	OCR exam reference	Python
FOR loop (Count controlled)	for x = 0 to 9 print("Hello") next x	for x in range(10): print("Hello")
	This will print the word "Hello" 10 times (0–9 inclusive)	
	for x = 2 to 10 step 2 print(x) next x	for x in range(2,12,2): print(x) # Python range function excludes end value

	This will print the even numbers from 2 to 10 inclusive	
	for x = 10 to 0 step -1 print(x) next x	for x in range(10,-1,-1): print(x) # Python range function excludes end value
	This will print the numbers from 10 to 0 inclusive	
WHILE loop (Condition controlled)	while colour != "Red" colour = input("New colour") endwhile	while colour != "Red": colour = input("New colour")
	This will loop until the user inputs the colour "Red". Check condition is carried out before entering loop	
DO WHILE loop (Condition controlled)	do colour = input("New colour") until answer == "Red"	# Not used in Python
	This will loop until the user inputs the colour "Red". Loop iterates once before a check is carried out	

Your notes

How to identify programming constructs

- You can identify which programming constructs are used by looking at certain **keywords**
- The keywords **if, elseif, else, endif, switch, case** indicate that the construct is **selection**
- The keywords **for, while, do** indicate that the construct is **iteration**
- If none of these keywords are used, this is a good indication that the construct is **sequence**

Worked Example

Tick (✓) one box in each row to identify whether each programming construct has or has not been used in the program **[3]**

```
total = 0

for i = 1 to 5

num = input("Enter a number")

total = total + num

next i

print(total)
```

	Has been used	Has not been used
Sequence		
Selection		
Iteration		

How to answer this question

- Always look for keywords first, if you find any then that construct must have been used
- Sequence must always be used if the program works

Answer

	Has been used	Has not been used
Sequence	✓	
Selection		✓
Iteration	✓	

Your notes

Common Arithmetic Operators

Your notes

Common Arithmetic Operators

- To demonstrate the use of common arithmetic operators, three sample programs written in Python are given below
- Comments have been included to help understand how the arithmetic operators are being used
  - **Arithmetic operators #1** - a simple program to calculate if a user entered number is odd or even
  - **Arithmetic operators #2** - a simple program to calculate the area of a circle from a user inputted radius
  - **Arithmetic operators #3** - a simple program that generates 5 maths questions based on user inputs and gives a score of how many were correctly answered at the end

Python code

```
# --------------------------------------------------------------------
# Arithmetic operators #1
# --------------------------------------------------------------------
# Get the user to input a number
user_input = int(input("Enter a number: "))

# if the remainder of the number divided by 2 is 0, the number is even
if user_input % 2 == 0:
   print("The number is even.")
else:
   print("The number is odd.")

# --------------------------------------------------------------------
# Arithmetic operators #2
# --------------------------------------------------------------------
# Get the radius from the user
radius = float(input("Enter the radius of the circle: "))

# Calculate the area of the circle
area = 3.14159 * radius ** 2

# Display the calculated area
print("The area of the circle with radius",radius,"is",area)

# --------------------------------------------------------------------
# Arithmetic operators #3
# --------------------------------------------------------------------
```

```
# Set the score to 0
score = 0

# Loop 5 times
for x in range(5):
    num1 = int(input("Enter the first number: "))
    operator = input("Enter the operator (+, -, *): ")
    num2 = int(input("Enter the second number: "))
    user_answer = int(input("What is "+str(num1)+str(operator)+str(num2)+"? "))

# Check the answer and update the score
    if operator == '+':
        correct_answer = num1 + num2
    elif operator == '-':
        correct_answer = num1 - num2
    elif operator == '*':
        correct_answer = num1 * num2

    if user_answer == correct_answer:
        score = score + 1
    else:
      print("Sorry that's incorrect.")

print("Your score is:", score)
```


Your notes

Common Boolean Operators

Your notes

Common Boolean Operators

- To demonstrate the use of common Boolean operators, three sample programs written in Python are given below
- Comments have been included to help understand how the Boolean operators are being used
  - **Common Boolean operators #1** - a simple program that assigns Boolean values to two variables and outputs basic comparisons
  - **Common Boolean operators #2** - a simple program to output a grade based on a users score
  - **Common Boolean operators #3** - a simple program reads a text files and searches for an inputted score

Python code

```
# ------------------------------------------------------------
# Common Boolean operators #1
# ------------------------------------------------------------
# Assign a Boolean value to a and b
a = True
b = False

# print the result of a and b
print("a and b:", a and b)
# print the result of a or b
print("a or b:", a or b)
# print the result of not a
print("not a:", not a)

# ------------------------------------------------------------
# Common Boolean operators #2
# ------------------------------------------------------------

# Take input for the score from the user
score = int(input("Enter the score: "))

# Compare the score and output the corresponding grade
if score >= 90 and score <= 100:
  print("Grade: A")
elif score >= 80 and score < 90:
  print("Grade: B")
```

```
elif score >= 70 and score < 80:
  print("Grade: C")
elif score < 70:
  print("Fail")

# ----------------------------------------------------------
# Common Boolean operators #3
# ----------------------------------------------------------
# Open the file for reading
file = open("scores.txt","r")
# Set flags to false
end_of_file = False
found = False
score = input("Enter a score: ")
# While it's not the end of the file and the score has not been found
while not end_of_file and not found:
  # read the line
  scores = file.readline().strip()
  # if the line equals the score
  if score == str(scores):
    found = True
    print("Score found")
  # if the line is empty
  if scores == "":
    end_of_file = True
    print("Score not found")
file.close()
```

Your notes

Data Types


Your notes

Primitive Data Types

What is a data type?

- A data type is **a classification of data into groups** according to the **kind of data they represent**
- Computers use **different data types** to represent **different types of data** in a program
- The basic data types include:

Data type	Used for	Example
Integer	Whole numbers	10, -5, 0
Real	Numbers with a fractional part	3.14, -2.5, 0.0
Character	Single character	'a', 'B', '6', '£'
String	Sequence of characters	"Hello world", "ABC", "@#!%"
Boolean	True or false values	True, False

- It is important to choose the correct data type for a given situation to ensure **accuracy** and **efficiency** in the program
- Data types can be changed within a program, this is called **casting**

What is casting?

- Casting is when you **convert one data type** to **another data type**

Example

- The following Python program is used to capture a users age to determine if they are old enough to vote

Line	Python code
01	age = input("Enter age")

02	if age >= 18:
03	print("Old enough to vote")
04	else:
05	print("Too young to vote")

- In this example, on line 01, no specific data type is requested
- By default the data type is stored as 'string'
- On line 02, a run-time error would occur because age is stored as a string and is being compared to an integer value in the selection statement
- Casting the age from a string to an integer would solve the error

Line	Python code
01	age = input("Enter age")
02	if int(age) >= 18:
03	print("Old enough to vote")
04	else:
05	print("Too young to vote")

- In the corrected code, casting is highlighted in green

Casting between data types

Conversion	Example	Output
From **Integer** to **Real**	int_value = 5 real_value = float(int_value)	5.0

Your notes

From **Real** to **Integer**	real_value = 5.7 int_value = int(real_value)	5
From **String** to **Integer**	str_value = "10" int_value = int(str_value)	10
From **Integer** to **String**	int_value = 5 str_value = str(int_value)	"5"
From **Boolean** to **String**	bool_val = True str_val = str(bool_val)	"True"
From **String** to **Boolean**	str_value = "True" bool_val = bool(str_value)	True

Your notes

Worked Example

Customers booking a holiday can choose between half board or all inclusive and a hotel star rating between 1 and 5

A typical booking record is shown in the table:

firstName	Jacob
lastName	Franks
boardType	All inclusive
starRating	5
bookingComplete	True

State the most appropriate data type for the following fields **[2]**:

boardType	

starRating	

Give the name of **one** field that could be stored as a Boolean data type **[1]**

Answer

boardType	**String**
starRating	**Integer**

- bookingComplete


Your notes