

تم تحميل هذا الملف من موقع المناهج الإماراتية



الملف حل الوحدة الأولى الخوارزميات والبرمجة

[موقع المناهج](#) ← [المناهج الإماراتية](#) ← [الصف الثامن](#) ← [تصميم](#) ← [الفصل الأول](#)

روابط مواقع التواصل الاجتماعي بحسب الصف الثامن



روابط مواد الصف الثامن على تلغرام

[الرياضيات](#)

[اللغة الانجليزية](#)

[اللغة العربية](#)

[التربية الاسلامية](#)

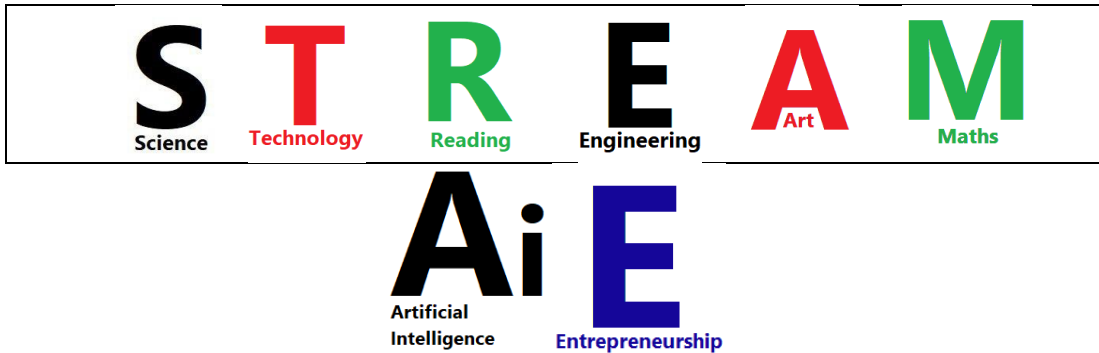
المزيد من الملفات بحسب الصف الثامن والمادة تصميم في الفصل الأول

ملف بوضح طبعة الأسئلة الوزارية لإمتحان مادة التصميم والتكنولوجيا من الى	1
نموذج إمتحان للصفوف السادس والسابع والثامن والتاسع	2
حل كتاب الطالب	3
بوب كويز الوحدة الأولى والثانية	4
بوب كويز الوحدة الثالثة	5

Table of Contents

Table of Contents	1
Unit 1: Algorithms and Programming	2
E-safety	3
Impacts of computing technologies	3
Decomposition	4
Algorithms	7
Flowcharts	7
Pseudocode	9
Variables	10
Data types	11
Changing data types	11
Advanced selection and repetition	15
Conditional statements	15
Selection	17
Iteration	24
Functions and methods	27
Comments and attribution	28
Using feedback	30
Unit 1 summary	32
Student reflection	32
End of unit quiz	33

Unit 1: Algorithms and Programming



E-safety

Impacts of computing technologies



Activity 1

Read the sentences about computing technologies. Decide if they describe positive or negative impacts.

Sentences	Effect [✓]	
	Positive	Negative
You can work faster and get better results with tools, such as an office package.	✓	
You can get fast access to information using the world wide web.	✓	
Not all information available with technology is accurate.		✓
Your verbal communication may not develop if you always use technology.		✓
You can communicate with people around the world using telephones, email, and social media.	✓	
You can have health problems, such as eye strain, muscular problems and sleeping when using technology.		✓
You can design and create new technology using programming skills and electrical components.	✓	
You can be isolated from family and friends when using technology.		✓

How can you get a good balance between computing technologies, everyday activities and your career options?

No teacher answer provided for open ended questions.

Decomposition



Activity 2

To complete this activity, you will apply computational thinking skills to help plan a computer game. The computer game involves moving a character around a simple world.



Remember, computational thinking skills include decomposition, abstraction and algorithmic thinking.

Decomposition: break down the problem

The character in the game needs to be able to perform certain actions, and each action needs to be handled differently by the game.

One of the requirements is that the player needs to be able to move using the commands forward, turn left, and turn right. The player should be able to pick up, use, and drop an item. The player should also be able to cast a spell and learn new spells.

The actions the player can perform have started to be broken down into smaller problems so that they are easier to solve. Complete the table for the required player actions.

Move	Item	Spell
forward	pick up	cast a spell
turn left	use	learn new spells
Turn right	drop	

Abstraction: remove unnecessary information

A graphic artist has designed part of the town that the player can explore, shown below.



The programmers want to test the movement of the player around this part of the town. They do not need all of the town details, just the buildings and paths.

Sketch a simple version of the map with all unnecessary information removed in the table below.

The sketch should be a simple representation including buildings and paths.

--

Algorithmic thinking: create a simple step-by-step algorithm

The command forward can be used to move the character forward in the direction that it is currently facing, and the character will stop moving forward if it reaches a split in the path, or a door.

The commands turn left and turn right can also be used to change the direction relative to the character; for example, if the character is facing to the left, the command turn right will result in the character facing up.



Write down the order of the movements that a game character could follow to go from a large building to a small house, travelling along the path.

No.	Instruction
1.	move forward
2.	turn left
3.	move forward
4.	turn right
5.	move forward
6.	turn right
7.	move forward
8.	turn right
9.	move forward
10.	
11.	
12.	
13.	
14.	
15.	

Algorithms

Flowcharts



Activity 3

Create a flowchart algorithm that simulates rolling two dice, then checks whether the values are the same (a double). The instructions for the algorithm are:

- generate a random number between 1 and 6 to simulate the first die roll.
- generate a random number between 1 and 6 to simulate the second die roll.
- compare the two dice values.
- if the values are the same, tell the user they rolled a double.
- else, tell the user they did not roll a double.

No teacher answers provided for open ended activities.



Activity 4

Create a flowchart algorithm for a coin toss game. The instructions for the game are:

- ask the user to guess heads or tails.
- simulate a coin flip by randomly generating either 'heads' or 'tails'.
- compare the coin flip with the user's choice.
- if the coin flip is the same as the user's choice, tell the user they won.
- else, tell the user they lost.

No teacher answers provided for open ended activities.

المنهج الإلكتروني

Pseudocode



Activity 5

Create a pseudocode algorithm for a login program where user user will enter a password. In the program a user can make multiple attempts to enter the correct password.

You can use the text-based example program for reference to help create a pseudocode algorithm:

```
1 stored_pass = "password"
2 password = ""
3 pass_mismatch = stored_pass != password
4
5 while pass_mismatch:
6     print("Enter your password:")
7     password = input()
8     pass_mismatch = stored_pass != password
9
10 print("Access granted")
```

Use the table below to create a pseudocode algorithm.

No.	Pseudocode instruction
1.	Open ended activity. One possible answer included below:
2.	Stored pass = "password"
3.	Password = ""
4.	While password != stored_pass
5.	Print "Enter your password:"
6.	Password = user input
7.	Print "Access granted"
8.	
9.	
10.	
11.	
12.	
13.	
14.	
15.	

Variables



Activity 6

Demonstrate your understanding of variables.

1. Identify variable names and data stored. Then, complete the table below:

Assignment statement	Variable name	Data stored in the variable
number = 2048	number	2048
score = 2.718	score	2.718
yearBorn = 2007	yearBorn	2007
thisYear = 2021	thisYear	2021
age = thisYear - yearBorn	age	14

2. Use programming software to create and assign values for the variables in the table below.

Scenario	Assignment statement	Data stored in the variable
1. A car weighs 3400 kg. How would you store the weight in a variable?	carWeight = 3400	3400
2. Ibrahim had three lives at the start of a game. He has just lost one life.	Lives = 2	2
3. Store the total number of Grade 9 students in a school. There are 43 students.	G9Students = 43	43

Remember, you must give variables appropriate names and assign values using an "=" operator. A print() command can be used to output variable values.

Data types

Changing data types



Activity 7

Demonstrate your understanding of variables.

1. Match the sample data with the correct data types.

Sample data	Data type
2000 int	int
"6c" string	string
"R" char	char
-1.4 float	float

2. Identify the data type needed to store the sample data.

Write your answers below.

Sample data	Data type
"Hello"	string
55	int
0.009	float
"Y"	char

2. Create and execute the program above using programming software, then answer the questions below.

3. Was your prediction correct or did something unexpected happen?

Write your answer below.

No teacher answers provided for open ended questions.

4. Complete your program investigation by answering the questions below

Questions	Answers
What data type is being collected at line 2?	String
Run the program and type more than 1 character when asked for your first initial. What happens?	More than 1 character is accepted as input.
Run the program and type in some string (text) when you are asked for your age. What happens and why?	String text is accepted as input
Line 12 contains the following code "likes_robot == robot". What is happening here?	Conditional statement checking if the values of two variables are equal
Run the program and type 9 when you are asked for your age. What happens?	9 is accepted as input, if the calculation was done properly the result would be 0.9 decades.
On line 10, the forward slash / is being used between the variable age and the number 10. What arithmetic operation is being performed?	Division

On line 10, change the word float to int. Run the code and type in 9 as your age. What happens?	Program still fails as data conversion must be done before the calculation.
Keeping line 10 as int. Run the code and type in 28 as your age. What happens?	Program still fails as data conversion must be done before the calculation.
What is happening when the number is stored as an integer instead of a float?	Number cannot be converted from string to float or integer in this way.

5. Can you change the program to improve the functionality?

Use the space below to plan the improvements to the program.

Teacher Answer

```

print("What is your first initial?")
initial = input()
print("What is your surname?")
surname = input()
print("What is your age?")
age = int(input())
print("True or False - are you a robot?")
likes_robot = input()
robot = "True"
decades = age/10
print(f"well hello {initial} {surname}")
print(f"It is {likes_robot == robot} that you are a robot.")
print(f"This is probably because you are {decades} decades
old")

```

}

Advanced selection and repetition

Conditional statements



Activity 9

Demonstrate your understanding of conditional statements and relational operators.

Create conditional statements for each case description in the table below. Then, state if the result of each statement is True or False.

Case description	Conditional statement	True or False
Is a kbyte equal to 12 bits?	kbyte == 12	False
Is a kbyte less than 500 bytes?	kbyte < 500	True
Is a mbyte not equal to 0?	mbyte != 0	True
Is a gbyte equal to a mbyte ?	gbyte == mbyte	False
Is a mbyte less than 500 bytes?	mbyte < 500	False
Is a gbyte greater than a mbyte ?	gbyte > mbyte	True

Did you know?

Computer memory can be explained using binary bits, bytes, kilo-bytes, mega-bytes, giga-bytes and tera-bytes. Each unit of storage is greater than the last.



Activity 10

Demonstrate your understanding of conditional statements and relational operators.

1. Examine the conditional statements in the table below. Then, re-write each statement as a question.

Conditional statement	Question
$\text{distAbudhabi} > 3$	Is the distance to Abu Dhabi greater than 3?
$\text{distFujairah} < \text{distAlmirfa}$	Is the distance to Fujairah less than the distance to Al Mirfa?
$\text{distAbudhabi} == 314$	Is the distance to Abu Dhabi equal to 314?
$\text{distFujairah} != \text{distAlmirfa}$	Is the distance to Fujairah not equal to the distance to Al Mirfa?

2. State if the result of each statement is True or False using the following values for the variables:

$\text{distFujairah} = 111$

$\text{distAbudhabi} = 181$

$\text{distAlmirfa} = 300$

Conditional statement	True or False
$\text{distAbudhabi} > 3$	True
$\text{distFujairah} < \text{distAlmirfa}$	True
$\text{distAbudhabi} == 314$	False
$\text{distFujairah} != \text{distAlmirfa}$	True

Selection



Activity 11

Create a program using programming software to demonstrate selection. The program will be used to check if a student is in Cycle 2.

The program will:

- input a grade and store it in a variable
- convert the grade to a numeric data type
- use selection with conditional statements to check if the grade is in Cycle 2
- output whether the Grade is in cycle 2.

Use the space below to plan the conditional statements needed for the program.

Teacher Answers

```
grade = input("Enter a grade ")
grade = int(grade)
if grade > 4 and grade <9:
    print("Grade is in cycle 2")
else:
    print("Grade is not in cycle 2")
}
```



Activity 12

Create a program using programming software to demonstrate selection. The program will be used to check if a time is in the morning (am) or afternoon (pm).

The program will:

- input a time and store it in a variable.
- convert the time to a numeric data type.
- use selection with conditional statements to check if the time is in the morning.
- output whether the time is in the morning or afternoon.
- format the output for example "The time is 11am" or "The time is 12pm"

Use the space below to plan the conditional statements needed for the program.

Teacher Answers

```
time = input("Enter a time ")
time = int(time)
print(time)
print(str(time))
if time <12:
    print("The time is " + str(time) + "am")
else:
    print("The time is " + str(time) + "pm")
}
```



Activity 13

Demonstrate your programming knowledge by completing a program investigation. First, review the program below. Then, answer the questions below.

```
1 #Program to check if it is morning time
2 hour = input ("Please enter the hour of the day(24-hour clock):")
3 hour = int (hour)
4
5 if hour < 12:
6     print ("The time ", hour, "is am")
7 else:
8     print ("The time", hour, "is pm")
```

1. Explain what you think will happen when this program is executed?

Program inputs a time, that is converted to an integer. The program uses selection to output whether the time is in the morning (am) or afternoon (pm).

What will happen if the user input is '18'?

Program output will be: "The time 18 pm"

What will happen if the user input is '5'?

Program output will be: "The time 5 am"

What will the program output be if the user input is '12'?

Program output will be: "The time 12 pm"



Activity 14

Demonstrate your programming knowledge by completing a program investigation.

Use tracing to determine the output based on each input. Then, complete the tables below.

Program 1

```
1 print("Guess a number between 1 and 10:")
2 number = int(input())
3 if number == 7:
4     print("You got it!")
5 elif number < 7:
6     print("Higher")
7 else:
8     print("Lower")
```

Input	Output
7	you got it!
9	Lower
3	Higher

Program 2

```
1 print("Enter a number: ")
2 odd_even = int(input())
3 odd_even = odd_even % 2
4 if odd_even == 1:
5     print("Your number is odd")
6 elif odd_even == 0:
7     print("Your number is even")
```

Input	Output
67	Your number is odd
24	Your number is even
93	Your number is odd



Activity 15

Part 1

Create a flowchart algorithm for a UAE quiz game. The instructions for the game are:

- output each question:
 - country name?
 - independence year?
 - founder name?
 - capital?
 - national animal?
- input each answer and store it in a variable.
- use selection with conditional statements to check if each answer is correct.
- store a score in a variable.
- increase the score by 1 when an answer is correct.
- output the score at the end of the game.

No teacher answer provided for open ended questions.



Part 2

Create a UAE quiz game program using programming software.

The program will:

- output each question:
 - country name?
 - independence year?
 - founder name?
 - capital?
 - national animal?
- input each answer and store it in a variable.
- use selection with conditional statements to check if each answer is correct.
- store a score in a variable.
- increase the score by 1 when an answer is correct.
- output the score at the end of the game.

Teacher Answers

```
score = 0
print("UAE quiz game program")
country = input("country name? ")
independence = int(input("independence year? "))
founder = input("founder name? ")
capital = input("capital? ")
animal = input("national animal? ")

if country == "UAE":
    score = score + 1
if independence == 1971:
    score = score + 1
if founder == "Sheikh Zayed":
    score = score + 1
if capital == "Abu Dhabi":
    score = score + 1
if animal == "falcon":
    score = score + 1

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


Iteration



Activity 16

Create a program using programming software to demonstrate iteration. The program will be used to increment a counter.

The program will:

- input a number and store it in a variable.
- convert the number to a numeric data type.
- use iteration to increment a counter from 1 to the input number.
- output the value of the counter for each iteration.

Use the space below to plan the iteration needed for the program.

Teacher Answers

```
number = input("Enter a number ")
number = int(number)
for x in range(1, number+1, 1):
    print(x)
}
```



Activity 17

Demonstrate your understanding of iteration.

1. Use tracing to determine the program output based on different loop ranges. Then, complete the table below.

1. `for x in range(0, 5, 1):`


2. `print(x)`

Loop range	Output
<code>range(0, 5, 1)</code>	0, 1, 2, 3, 4
<code>range(0, 15, 3)</code>	0, 3, 6, 9, 12
<code>range(-20, 5, 2)</code>	-20, -18, -16, -14, -12, -10, -8, -6, -4, -2, 0, 2, 4

2. Plan the loop range required for the program to produce the following outputs. Then, complete the table below.

Output	Loop range
-1, 0, 1, 2, 3	<code>range(-1, 4, 1)</code>
0, 5, 10, 15, 25	<code>range(0, 30, 5)</code>



	Activity 18
---	--------------------

Demonstrate your understanding of iteration.

Identify the type of loop to use for each case description in the table below.

Case description	Loop type	
	while	for
Asking the user to enter their email until it is a valid email.	✓	
Printing all even numbers from 1 to 5000.		✓
Calculating your final score by adding up your grades in all the ten subjects you study.		✓
Showing a menu to a user where they can select an item or exit the program.	✓	
Continuing to heat water until the temperature sensor says it has reached the boiling point.	✓	
Generating a report showing all students who have signed up for a trip out of the 2,000 students in the school.		✓
Reading lines from a file until we reach the end of the file.	✓	
Continuing to play video game until you choose to exit.	✓	
Adding up all the money you spent on food this week.	✓	

Functions and methods



Activity 19

Create a program using programming software to demonstrate functions. The program will use a function to calculate and return a sum total of two numbers.

The program will:

- use a variable to store a total
- call a function that will:
 - input two numbers and store them in variables.
 - convert the numbers to a numeric data type.
 - calculate the sum total and return the value to the program.
- output the sum total

Use the space below to plan the function needed for the program.

Teacher Answers

```
def sum_total():  
    num_1 = int(input("Enter first number: "))  
    num_2 = int(input("Enter second number: "))  
    return num_1 + num_2  
  
total = sum_total()  
print(total)
```

```
}
```

Comments and attribution



Activity 20

Demonstrate how to add programming comments and give attribution to others by adding the following comments to the example program.

Comments
Programming comments are important
They make your code easier to understand and maintain
This is "maths.py" which demonstrates some basic maths in Python
First, we will ask the user to input two numbers.
Then, we do some calculations
Most computer languages use the asterisk for multiplication
and the forward slash for divide
Now output the result
Here I used an ASCII code to output the traditional divide symbol

Example program

```
1 #
2 #
3 #
4
5 print("This program does some simple maths.")
6
7 #
8 a = int(input("Please enter your first number: "))
9 b = int(input("Now give me your second number: "))
10
11 #
12 add = a + b
13 minus = a - b
14 times = a * b #
15 Divide = a / b #
16
17 #
18 print(a, "+", b, "=", add)
19 print(a, "-", b, "=", minus)
20 print(a, "x", b, "=", times)
21 print(a, chr(247), b, "=", divide) #
```

Teacher Answers

Comments should be included in appropriate places to explain the program.

```
#Programming comments are important
#They make your code easier to understand and maintain
#This is "maths.py" which demonstrates some basic maths in
Python

print("This program does some simple maths.")

# First, we will ask the user to input two numbers.
a = int(input("Please enter your first number: "))
b = int(input("Now give me your second number: "))

# Then, we do some calculations
add = a + b
minus = a - b
times = a * b #Most computer languages use the asterisk for
multiplication
divide = a / b #and the forward slash for divide

#Now output the result
print (a, "+", b, "=", add)
print (a, "-", b, "=", minus)
print (a, "x", b, "=", times)
print (a, chr(247), b, "=", divide) # Here I used an ASCII
code to output the traditional divide symbol
```

```
}
```

Using feedback



Activity 21

Demonstrate how to improve a program by using feedback. First, work in teams to review the program below. Then, answer the questions below.

```
1 def password_check():
2     password = "12345"
3     entered_pass = ""
4     pass_not_valid = password != entered_pass
5     attempts = 0
6     while pass_not_valid and attempts < 3:
7         print("Enter a password:")
8         entered_pass = input()
9         attempts = attempts + 1
10        if password == entered_pass:
11            break
12        if password != entered_pass:
13            return "Access Denied"
14        else:
15            return "Access Granted"
16
17
18 print(password_check())
```

1. Answer the questions to collect team feedback on the program:

- What is the purpose of the program?
Login or Password checking program.
- Does the program work and is it readable?
Yes the program works, however it is not very human readable.
- What programming constructs have been used?
Selection, Iteration (Also, a function for program organisation).
- Have comments been included to explain the program?
No
- How could you improve the program?
Remove unnecessary code, optimise remaining code, improve human readability of the conditional statements

2. Create and execute the program above using programming software. Then, use the feedback on the program to identify and make some improvements.

Use the space below to plan the improvements to the program.

No teacher answers provided for open ended activities.

One example program included below:

```
def password_check():
    password = "12345"
    entered_pass = ""
    attempts = 0
    while attempts < 3:
        print("Enter a password:")
        entered_pass = input()
        attempts = attempts + 1
        if password == entered_pass:
            return "Access Granted"
    return ("3 incorrect attempts, Access Denied")

print(password_check())
```


Unit 1 summary

Student reflection






Activity 22

Write about what you have learned and liked in this unit.

No teachers answers provided for open ended activities.

Three things I have learned: 1. _____ 2. _____ 3. _____
Two things I have liked: 1. _____ 2. _____

Knowledge and skills reflection

Please tick the box to show what you understand:			
I prepared a solution from the decomposed parts of a problem to facilitate design, implementation, and review of a program			
I create pseudocode and flowcharts to represent algorithms for complex programming problems.			
I constructed programs using operations to change the value of appropriately names variables.			
I designed and iteratively developed programs using loops and compound conditionals.			
I construct an organised program by reusing procedures with different data inputs.			
I analysed feedback from team members to refine a program and ensure user needs are met.			
Teacher's comments:			



Activity 23

What have you learned? Do the quiz.

1. Some methods used to create algorithms are pseudocode and _____.

- | | |
|---------------|----------------|
| A. pie charts | B. flowcharts |
| C. bar charts | D. line graphs |

flowcharts

2. Basic data storage in computer programs is done using _____ and constants.

- | | |
|-------------|--------------|
| A. variants | B. variables |
| C. vectors | D. visuals |

variables

3. A function is a group of indented commands with a name used to do a specific task in a program.

- | | |
|---------|----------|
| A. True | B. False |
|---------|----------|

True

4. Functions are commonly used to perform _____ tasks such as calculations, data output and file handling.

- | | |
|------------|---------------|
| A. simple | B. short |
| C. unusual | D. repetitive |

repetitive

5. Using feedback is one way to identify improvements that can be made to computer programs.

- | | |
|---------|----------|
| A. True | B. False |
|---------|----------|

True