

Example 2.1 – Code readability

GCSE Computer Science (2016), SAM Paper 2, Q4(a)

- 4 A computer programmer uses a programming language to write program code for Sparky Autos.

(a) State **two** techniques that the programmer could use to make the code easy to read.

(2)

1

.....

2

.....

Example 2.2 – Code readability

iGCSE Computer Science, SAM Paper 2, Q1(b)

- (b) Open the file **Q01b** in the code editor.

Answer these questions about the code.

- (i) Identify **one** technique that could be used to make the code in **Q01b** more readable.

(1)

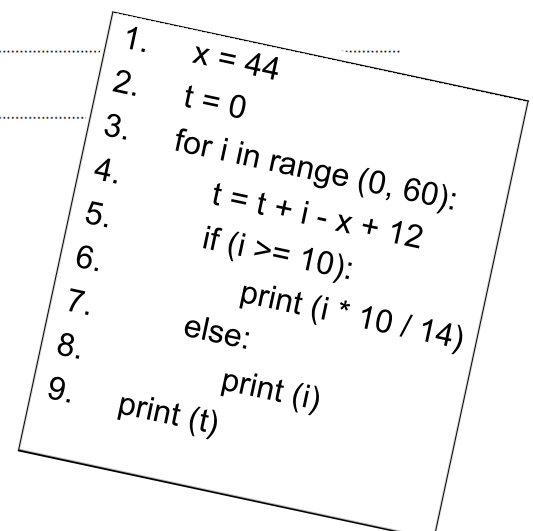
.....

- (ii) State **one** reason why code should be readable.

(1)

.....

.....



```
1. x = 44
2. t = 0
3. for i in range (0, 60):
4.     t = t + i - x + 12
5.     if (i >= 10):
6.         print (i * 10 / 14)
7.     else:
8.         print (i)
9. print (t)
```

Example 2.3 – Types of error

iGCSE Computer Science, SAM Paper 2, Q1(c)

- (c) (i) Give a definition of a syntax error.

(1)

-
- (ii) Open the file **Q01c** in the code editor.

Amend the code to correct three program errors.

Save your amended code as **Q01cFINISHED** with the correct file extension for the programming language.

(3)

```
1. myNumbers = [10, 20, 30, 40 ,50, 60, 70, 80, 90, 100]
2. total
3. for theNumber in myNumbers:
4.     total = total + theNumber
5.     if (theNumber % 2 = 0):
6.         print("Even")
7.     else:
8.         print("Odd")
9. print(total)
```

Example 2.4 – Test data

GCSE Computer Science (2016), Specimen Paper 2, Q2(b)

- 2 The HappyPetBox Company offers discounts based on the type of box ordered and the length of the subscription selected.

```

1
2 IF ((subscriptionLength = 12) AND (boxType <> "Standard")) THEN
3     SEND "gold star discount" TO DISPLAY
4 ELSE
5     IF (subscriptionLen > 5) AND ((boxType = "Large") OR (boxType = "Medium")) THEN
6         SEND "silver star discount" TO DISPLAY
7     ELSE
8         IF ((subscriptionLen > 2) AND (boxType <> "Standard")) THEN
9             SEND "bronze star discount" TO DISPLAY
10        ELSE
11            SEND "regular pricing" TO DISPLAY
12        END IF
13    END IF
14 END IF
15
16

```

- (b) The algorithm needs to be tested thoroughly.

Complete the table to show boundary test data to meet the requirements.

(3)

Requirements	Input	
	subscriptionLen	boxType
A condition generating a bronze star discount		
Smallest subscription qualifying for a gold star discount		
Largest subscription qualifying for no discount		

Example 2.5

iGCSE Computer Science, SAM Paper 2, Q3(b)(vii)

(vii) One line in the code is identified as **not** working as expected.

State why this line does **not** work as expected.

(1)

```
1. import time
2.
3. theDate = ""
4.
5. def toCelsius (inTemp):
6.     celsius = 0
7.     celsius = (5.0 / 9.0) * (inTemp - 32.0)
8.
9. def toFahrenheit (inTemp):
10.    fahrenheit = ((9.0 / 5.0) * inTemp) + 32.0
11.    return (fahrenheit)
12.
13. def waitTenSeconds ():
14.    time.sleep (10)
15.
16. def waitTime (inSeconds):
17.    time.sleep (inSeconds)
18.
19. print (toFahrenheit (0))
20. print (toCelsius (212.0)) # This line does not work properly
21. print ("Sleeping for 10")
22. waitTenSeconds()
23. print (toCelsius (32.0))
24. print ("Sleeping for 5")
25. waitTime (5)
26. print (toFahrenheit (100.0))
```

The subprogram (being called on this line) is missing a return statement.

Example 2.6 – Trace table
iGCSE Computer Science, SAM Paper 2, Q5(a)

5 Data, stored as numbers, is very easily processed using computer algorithms.

(a) Open the file **Q05a** in the code editor.

Complete the trace table to show the execution of the code.

You may not need to fill in all the rows in the table.

(5)

target	rs	rm	r

```
target = 4
r = 1
rs = 0
rm = 0

while (r <= target):
    rs = r ** 2
    print(rs)
    rm = r % 4
    print(rm)
    r = r + 1
```

Example 2.7 – Structural components

iGCSE Computer Science, SAM Paper 2, Q1(a)

1 Programmers use accepted programming constructs when writing code.

(a) Open the file **Q01a** in the code editor.

Answer these questions about the code.

(i) Identify the name given to a **data structure** in the code.

(1)

(ii) Identify the line number(s) showing **repetition**.

(1)

(iii) Identify the line number(s) showing **selection**.

(1)

(iv) Identify the name of a **variable**.

(1)

```
1. myNumbers = [20, 30, 40, 50]
2. total = 0
3. for i in range (5, 15):
4.     total = total + i
5.     if (i < 10):
6.         print (i * 10)
7.     else:
8.         print (i)
9. print (total)
```

Example 2.8 – Data types and structures

iGCSE Computer Science, SAM Paper 2, Q6

6 Open the file named **Q06** in the code editor.

In file **Q06**, the names and years of birth of artists are stored in a 2-dimensional data structure.

Labels for their work need to be created by joining the first letter of their last name, the first letter of their first name and their year of birth.

For example, a label for ('Andy', 'Warhol', 1928) would be 'WA1928'.

Write a program to:

- process each artist to create a label
- store all the labels in the data structure named 'theLabels'
- display the labels for all the artists
- find and display the name and year of birth of the youngest artist.

Your program should function correctly, even if 'theArtists' data structure has more, fewer, or different artists.

You **must** use the data structures in file **Q06**.

Save your amended code as **Q06FINISHED** with the correct file extension for the programming language.

(Total for Question 6 = 20 marks)

```
theArtists = [
    ["Andy", "Warhol", 1928],
    ["Pablo", "Picasso", 1881],
    ["Salvador", "Dali", 1904],
    ["Lavinia", "Fontana", 1552],
    ["Jackson", "Pollock", 1912],
    ["Henri", "Matisse", 1869],
    ["Frida", "Kahlo", 1907],
    ["Georgia", "O'Keeffe", 1887],
    ["Kara", "Walker", 1969],
    ["Yayoi", "Kusama", 1929]
]

theLabels = []    # Put the new user labels into this structure

# Add your code here
```

Example 2.9 – Data types and structures

GCSE Computer Science (2016), SAM Paper 2, Q5(c)

- (c) Complete the table to give the appropriate data type of a variable to store each item.

(4)

Item	Data type
Gender of individual staff member	
Whether an individual car is still under the manufacturer's warranty	
Mean number of hours needed to recharge the battery in each car	
The number on the individual car	

- (d) Each member of staff:

- has a 4-digit ID number, such as 3865 or 4722
- works a whole number of hours on the days they work.
- works no more than 12 hours in a single day.

The business is open 7 days a week.

Draw a diagram of a data structure that shows the hours worked for each day of the week. Include data for at least **two** members of staff.

(3)

Example 2.10 - Validation

GCSE Computer Science(2016), Specimen Paper 2, Q6(a)

- 6 The HappyPetBox Company uses a software application to calculate staff wages and produce payslips. Sample input data for this system is shown.

National insurance (NI) number	Standard format LL123456L
Full time	Y or N Full-time = minimum of 40 hours per week Part-time = maximum of 20 hours per week
Hours worked	Integer Hours worked in current week Maximum of 10 hours overtime in one week for full-time only
Pay rate	Real Hourly pay rate

- (a) The input data needs to be validated.
- (i) Complete the validation rules in the pseudo-code.

Write your answers in the boxes provided.

(6)

```

3  # Validate NI number
4  STRING NInumber
5  INTEGER hoursWorked
6  CHARACTER fullTime
7
8  # Validate National Insurance number
9  IF LENGTH (NInumber)  THEN
10
11     SEND "Error message" TO DISPLAY
12 END IF
13
14 # Validate hours worked this week
15 IF hoursWorked < 0  THEN
16
17     SEND "Error message" TO DISPLAY
18 END IF
19
20 # Validate part-time worker's hours
21 IF fullTime = 'N'  THEN
22
23     SEND "Error message" TO DISPLAY
24 END IF

```

- (ii) Create a suitable 'error message' to explain one of your validation rules to the user.

(1)

Example 2.11 – Reading from and writing to files

iGCSE Computer Science, SAM Paper 2, Q3(c)

Open the code named **Q03c** in the code editor.

Write a program to implement these requirements.

For all lines in the **Cities.txt** file, the code must:

- read the line
- append a line number and a space to the front
- write the new line to a **Numbered.txt** file
- print the line to the display

You must use the structure given in file **Q03c** to complete the program.
Do not add further functionality.

Save your amended code as **Q03cFINISHED** with the correct file extension for the programming language.

(7)

```
# Do not use any other data structure such as an array or a list.
count = 0           # A counter for the line numbers
theLine = ""        # Holds each line of the file

# Open "Cities.txt" as input

# Open "Numbered.txt" as output

# Use a for/each loop to read each line of
# the input file into a variable named 'theLine'

    # Increment the line count

    # Add the line number to the front of the line followed by a space

    # print the line to the display

    # Write the new line to the output file

# Close the input file

# Close the output file
```

Example 2.12 – Logical operators

iGCSE Computer Science, SAM Paper 2, Q2(c)

```
myList = [(800, 23000), (1499, 10000), (1600, 47000), (200, 10000)]
for pair in myList:
    income = pair[1]
    attendance = pair[0]
    print ("Attendance: ", attendance, " income: ", income)

    if ( ):
        print ("Sufficient profit made this week")
    elif ( ):
        print ("income in line with attendance this week")
    elif ( ):
        print ("Attendance is very low this week. Contact the fan club.")
    else:
```

Condition	Output message
Attendance is at least 1500	Sufficient profit made this week
Income is at least 45000	Sufficient profit made this week
Attendance is at least 750; income is at least 22500	Income in line with attendance this week
Attendance is fewer than 500	Attendance is very low this week Contact fan club
All other inputs	Possible accounting error

(4)

Example 2.13 – Subprograms

GCSE Computer Science(2016), SAM Paper 2, Q3(a)and Q3(b)

- 3** A holiday company has a website. They would like to publish the daily temperatures in their most popular destinations.

(a) Describe **one** benefit of using subprograms.

(2)

- (b) The holiday company needs to be able to convert temperatures between Celsius and Fahrenheit.

Open the file **Q03b** in the code editor.

Answer these questions about the code.

- (i) Identify the name of a **built-in** subprogram in the code.

(1)

- (ii) Identify the name of a **user-defined** subprogram.

(1)

- (iii) Identify the name of **one** input parameter used in a subprogram.

(1)

- (iv) Identify the name of a subprogram that does **not** use input parameters.

(1)

- (v) Identify the name of a **local** variable.

(1)

- (vi) Identify the name of a **global** variable.

(1)

- (vii) One line in the code is identified as **not** working as expected.

State why this line does **not** work as expected.

(1)

Example 2.14 – Library modules

GCSE Computer Science, June 2016 Q3(c)(i)

Candidate 1

(c) Shaneela is planning to use libraries in the code for the website.

(i) State **two** reasons for using libraries in code.

(2)

1 Don't have to repeat pieces of frequently used long code.

2 The code is easier to understand.

Candidate 2

1 Using libraries removes the need to program

simple things that could be included in a library, saving time.

2 Libraries are usually without any errors and are compiled ~~already~~ ^{programs that are} and ~~executed~~ executed by the interpreter, so it will be more efficient to use libraries than ~~writing them~~ ^{writing them} ~~yourself~~ ^{yourself}.

Candidate 3

(2)

1 Lots of information/^{data} can be obtained from libraries.

2 The use of libraries in code speeds up the ~~pre~~-running speed of the program.

Example 2.15 – Generalising a subprogram

GCSE Computer Science, June 2016 Q4(c)(i)

- (c) Manuel is writing a program to help pupils with their maths revision.

Here is some pseudocode for an algorithm that calculates the area of a circle, with radius of 5 units. It uses a subprogram.

```
calcCircleArea ()  
  
PROCEDURE calcCircleArea ()  
BEGIN PROCEDURE  
    SET area TO Pi * 5 * 5  
    SEND area TO DISPLAY  
END PROCEDURE
```

Manuel wants to change the pseudocode so that the area of any circle can be calculated using calcCircleArea.

- (i) Write the changed pseudocode in the box below.

(3)

Candidate 1

SET radius TO VALUE

~~PRO~~

calcCircleArea()

PROCEDURE calcCircleArea(radius)

BEGIN PROCEDURE

SET area TO Pi * radius * radius

SEND area TO DISPLAY

END PROCEDURE

Candidate 2

calCircleArea()

PROCEDURE calCircleArea()

~~Be~~ BEGIN PROCEDURE

radius = int(input("what is the radius"))

SET area TO $\pi * \text{radius} * \text{radius}$

SEND area TO DISPLAY

END PROCEDURE

Candidate 3

calCircleArea()
user number = input("enter a number")

PROCEDURE calCircleArea()

BEGIN PROCEDURE

SET area to $\pi * \text{user number} * \text{user number}$

SEND area TO DISPLAY

END PROCEDURE.
