

Thomas White - Marks

Q1 a) Algorithm for tea: 16/17

b) Different drink selections - 5 /5

c) Storing data on a computer - 1 /3 **Total for Question 1 - 22/25**

Q2 Flowchart 22 /25 **Total for Question 2 - 22 /25**

Q3 a) Python code for 2) above 3/15

b) (Advanced) - 0/10 **Total for Question 3 - 3/25**

Q4 a) - An oval in turtle - 9/9

b) A 5 point star in turtle - 8/8

c) A spiral in turtle - 0/8 **Total for Question 4 - 17/25**

OVERALL TOTAL - 64%

Mark agreed

64%

Computational Thinking Test Oct 2017

Name Thomas White

Student number 151802

Programme Business Information Technology

Answer only the questions you can – if you cannot answer a question, move on to the next question. This test is not negatively marked, so you may wish to guess if unsure. Questions 1 and 2 to be answered on this paper, questions 3 and 4 to be answered using Python and saved (then uploaded via Moodle for staff).

1) You have been selected to program a new robot intended to create hot drinks. The robot is capable of following simple, tea and coffee-oriented commands precisely, but has no understanding either of the process, or the fundamental principles which underpin it (e.g. that a kettle requires power). The robot has access to the following items:

- Kettle (initially unplugged) ✓
- ~~Tea bags~~
- Jar of ground instant coffee
- 1L carton of milk – open?
- Unopened bag of sugar
- 1 metal tea spoon – for sugar
- 1 large mug
- Access to a sink for water and an electrical socket for power

a) You are required to give the robot the set of instructions (algorithm) necessary to successfully make a cup of milky coffee with 1 teaspoon of sugar. Each instruction should be on a new line. Ensure that your instructions are in a logical order and no steps are missed; while highly capable of following instructions, the robot cannot solve problems independently.

(17)

Move Kettle and its base near to the plug socket
Plug the Kettle base into the socket
Pickup the Kettle
Move the Kettle over to the sink
Turn the tap on
open the Kettle lid
Put the kettle under the running tap and wait 20
Seconds
Turn off the tap
Close the Kettle lid
move the kettle back over to the base
put it down on the base
Turn on the Kettle and wait 2 mins for it to boil
move the mug close to the Kettle
open the Jar of Coffee
pickup the teaspoon
Scoop up one teaspoon of Coffee and drop the Coffee
in the mug.

3

✓ 3

(PTO for more space)

put down the teaspoon.

Pickup the kettle

Fill the mug $\frac{4}{5}$ of the way with water from the kettle ✓2

put down the Kettle on the base

pick up the carton of milk

Fill the remainder of the mug with milk ✓1

put down the ~~kettle~~ Carton of milk

open the bag of sugar ✓1

pick up the teaspoon

Scoop up one teaspoon of sugar and drop it into the mug, ✓3

put the teaspoon into the mug and stir for 20 seconds.

Take the teaspoon out and put it in the sink. ✓

(I have no idea how to make tea or coffee as I drink neither!) !! +2

(16)

- b) The robot is, of course, capable of making many variations of hot drink. The user must be permitted to give information about their drink preferences to the robot before it is created.

would you like tea or coffee? (tea or coffee)
do you have milk with your drink? (yes or no)
if yes, how milky do you like your hot drink? (not very, some milk and very milky)
Do you have sugar with your hot drink? (yes, no) ✓5
if yes, how many teaspoons of sugar? (1, 2, 3, 4, 5)

(5)

- c) If the instructions were converted to code, explain briefly how this data may be stored.
What is the name given to a piece of data stored by a computer?

Boolean String ✓1
integer ?

(3)

2)

We have been given the requirements for a program which are:

Requirement 1 - Says "Hello" to the user at the start.

Requirement 2- Reads a series of numbers from the user, adding them together.

Requirement 3 - Stops when the user inputs -1 as a value.

Requirement 4 - Once the user elects to stop inputting numbers, outputs the sum of the numbers input.

Requirement 5 - Says "Goodbye" to the user.

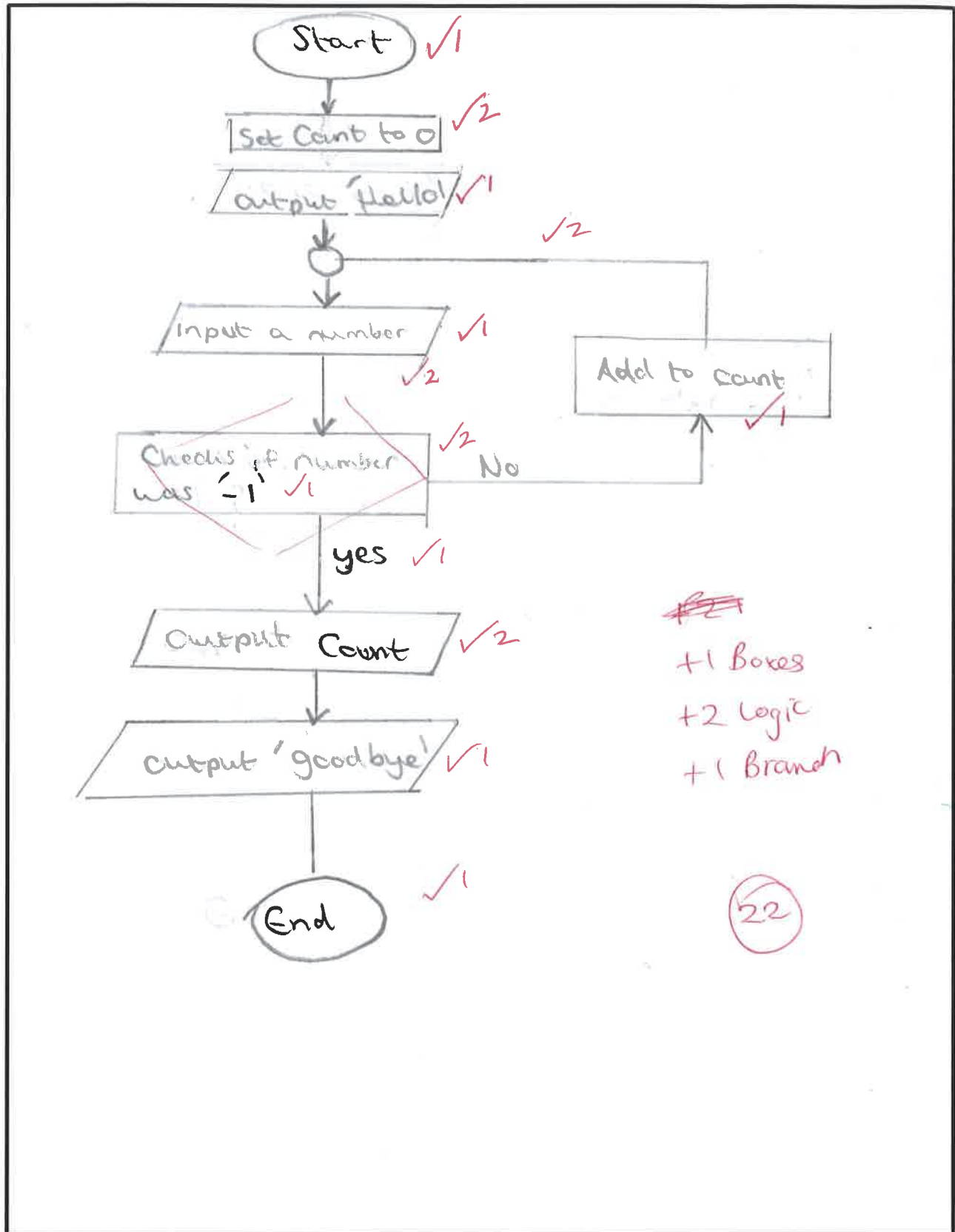
For example, the program may run something like:

```
Hello
Enter number:
8
Enter number:
10
Enter number:
3
Enter number:
-1
Your total is 21.
Goodbye.
```

Of course, the user may enter any set of numbers and get a valid result, here's another example of the program running:

```
Hello
Enter number:
9
Enter number:
-1
Your total is 9.
Goodbye.
```

Write the flowchart for this program:



- 3) Now write the in Python code for the program given in 2) above. Remember, the output might look similar to:

```
Hello.  
Enter number (-1 to exit): 8  
Enter number (-1 to exit): 10  
Enter number (-1 to exit): 3  
Enter number (-1 to exit): -1  
Your total is 21  
Goodbye.
```

(15)

- (Advanced) The program should additionally prevent the user from adding negative numbers (other than -1 to exit). For example:

```
Hello.  
Enter number (-1 to exit): 5  
Enter number (-1 to exit): 10  
Enter number (-1 to exit): -5  
You cannot input a negative number.  
Enter number (-1 to exit): 7  
Enter number (-1 to exit): 2  
Enter number (-1 to exit): 6  
Enter number (-1 to exit): -1  
Your total is 30  
Goodbye.
```

(10)

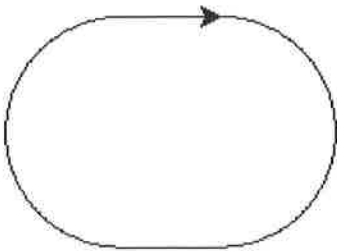
4) Using Python with turtle, draw the following shapes:

a) An oval. Remember that the code to draw a circle is:

```
import turtle

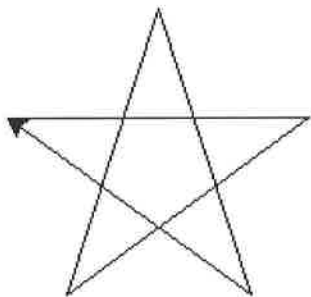
count = 0
while (count < 360):
    turtle.forward(1)
    turtle.right(1)
    count = count + 1
```

You should start with this as a base, and modify it to form an oval similar to:



(7)

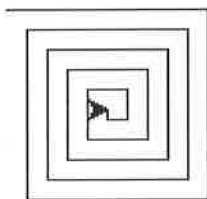
b) A 5 point star. This can be achieved without lifting the pen. It should look similar to:



Hint: the angle of rotation at the end of each point is 144 degrees.

(8)

c) A spiral. This may be a square spiral similar to:



You should not equate each edge to a line of code. Instead consider what you can use to reduce the amount of code you are required to produce.

(10)

Computational Thinking Test - 5 Pointed Star.py

#Thomas White

import turtle

turtle.forward(140)
turtle.right(144)

turtle.forward(140)
turtle.right(144)

turtle.forward(140)
turtle.right(144)

turtle.forward(140)
turtle.right(144)

turtle.forward(140)
turtle.right(144)

Computational Thinking Test - Oval.py

```
#Thomas white
import turtle

count = 0
while(count<180):
    turtle.forward(1)
    turtle.right(1)
    count=count+1

turtle.forward(80)

count1 = 0
while(count1<180):
    turtle.forward(1)
    turtle.right(1)
    count1=count1+1

turtle.forward(80)
```

Computational Thinking Test - Python.py

#Thomas White

print('Hello') ✓ 1

num = 1 ✓ 2 1

if num == -1: ✓
 print('goodbye') ✓
else

while number != -1

