

Christine Woods - Marks

Q1 a) Algorithm for tea: 15 /17

b) Different drink selections - 2/5

c) Storing data on a computer - 1/3 **Total for Question 1 - 18/25**

Q2 Flowchart 14 /25 **Total for Question 2 - 14 /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 - 52%



Mark agreed

52%

Computational Thinking Test Oct 2017

Name Christine Woods

Student number 1704446

Programme Degree (Part-Time)
Applied Computing

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
 - Unopened bag of sugar
 - 1 metal tea spoon
 - 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)

FILL KETTLE WITH WATER.
PLUG IN THE KETTLE TO ELECTRICAL SOCKET.
SWITCH ON THE KETTLE

GET THE MUG

OPEN THE JAR OF COFFEE

USE METAL TSPON TO PUT 1
SPOON OF COFFEE INTO MUG

OPEN THE BAG OF SUGAR ✓1

PUT 1 TEASPOON OF SUGAR INTO MUG ✓2

OPEN CARTON OF MILK

POUR MILK INTO MUG HALF ^{FILL} MUG ✓2

WAIT FOR KETTLE TO BOIL

POUR WATER FROM KETTLE INTO MUG UNTIL FULL ✓1

USE TEASPOON TO STIR COFFEE
✓1

15

(PTO for more space)

- 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.

COFFEE SHOULD CONSIST OF A HEAPED T-SPOON OF COFFEE
SUGAR SHOULD BE A HEAPED T-SPOON OF SUGAR.
MILK SHOULD HALF FILL THE MUG ✓
WATER MUST BE BOILING ✓
COFFEE MUST BE STIRRED. (2)

(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?

Data could be stored in a file. The instructions
if coded could be kept in a document within a
file location on the hard drive. (1)

(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.

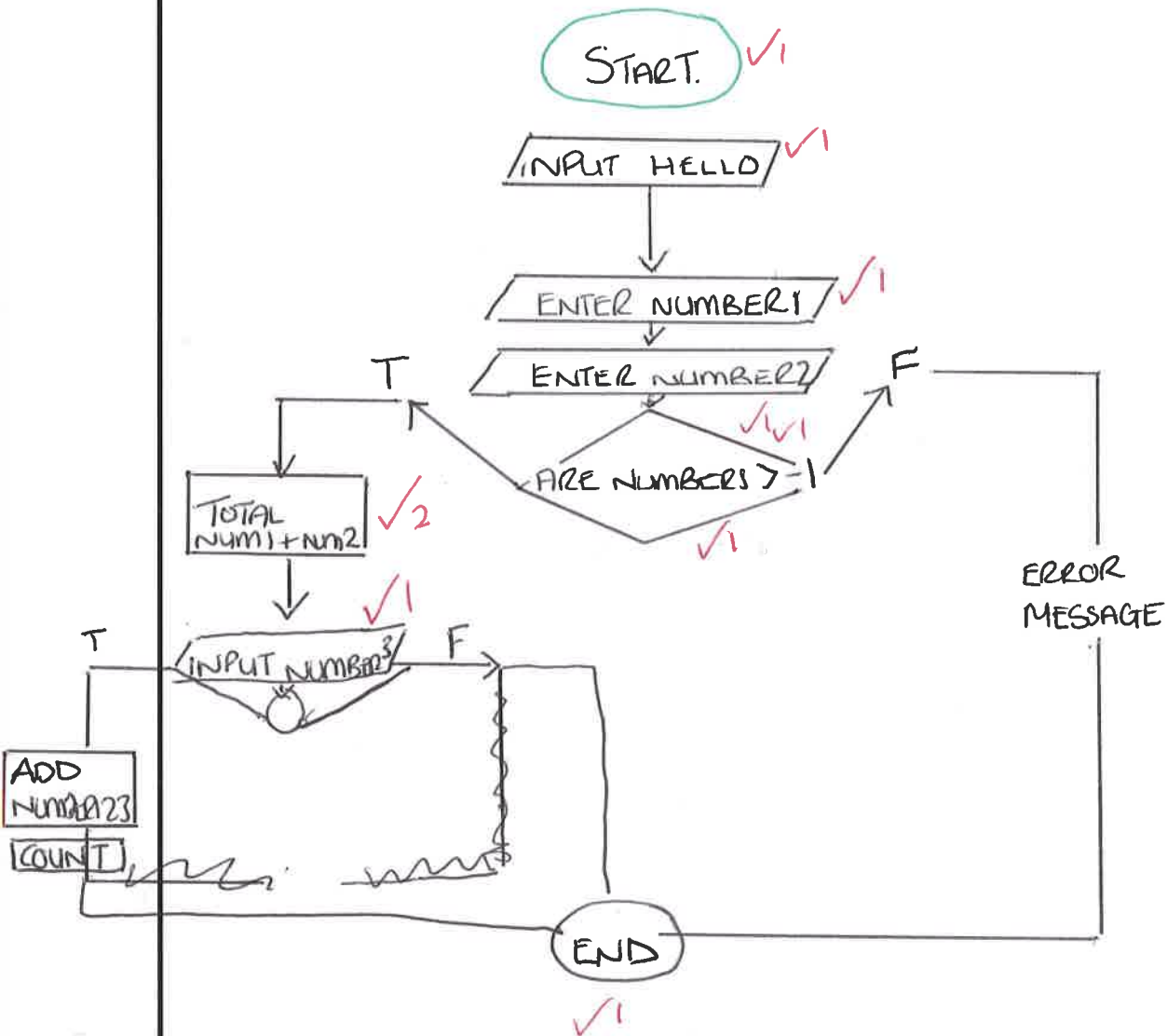
For example, the program may run something like:

```
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:



+ 2 Boxes
+ 1 Branch
+ 1 Logic

14

- 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)

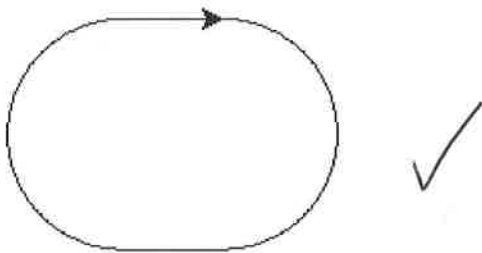
4) Using Python with turtle, draw the following shapes:

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

```
import turtle

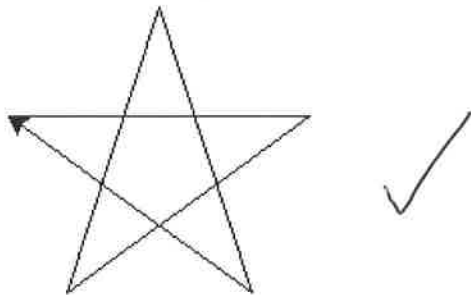
count = 0
while (count < 360):
    turtle.forward(1)
    turtle.right(1)
    count = count + 1
```

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



(7)

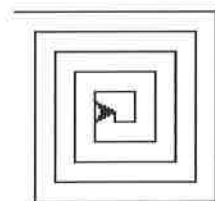
b) A 5 point star. This can be achieved without lifting the pen. It should look similar to:



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)

#Christine Woods

#question3

```
print("Hello") ✓  
num1=int(input("Enter the first number"))#this is for the first number  
num2=int(input("Enter another number")) #this is to enter the 2nd number  
total=num1+num2#total of the two numbers  
print ("The first number is {0}. The second number is {1}. The total is  
{2}".format(num1,num2,total))  
if (num1 or num2) <-1: #trying to get it to recognise that it cant be -1  
    print("invalid number")
```

✓ 3

No loop. Some output or.

```
#Christine Woodsimport turtle
```

```
turtle.forward(150)
```

```
count=0
```

```
while(count<180):
```

```
    turtle.forward(1)
```

```
    turtle.right(1)
```

```
    turtle.forward(1)
```

```
    count=count+1
```

```
turtle.forward(150)
```

```
count=0
```

```
while(count<180):
```

```
    turtle.forward(1)
```

```
    turtle.right(1)
```

```
    turtle.forward(1)
```

```
    count=count+1
```

Turtle Star.py

#Christine Woods

```
import turtle

turtle.forward(150)
turtle.right(144)
turtle.forward(150)
turtle.right(144)
turtle.forward(150)
turtle.right(144)
turtle.forward(150)
turtle.right(144)
turtle.forward(150)
```

