Task 2: software design and development

Scotven offers a mobile wi-fi service at outdoor events. They check the signal strength by taking readings from five locations.

The analysis and design for a program is shown below:

Program analysis

A program is required to display the five readings taken at the event and a signal pattern. The signal pattern will show the strength of readings (S = strong, M = medium and P = poor) in the order the readings were taken (1 to 5), for example "SSMPS".

Assumptions

- readings of signal strengths are recorded with two decimal places from 0.00% to 100.00%
- a strong signal is greater than 80% signal strength
- a medium signal is less than a strong signal and more than a poor signal
- a poor signal is less than 30% signal strength

Inputs

- five valid readings from the venue

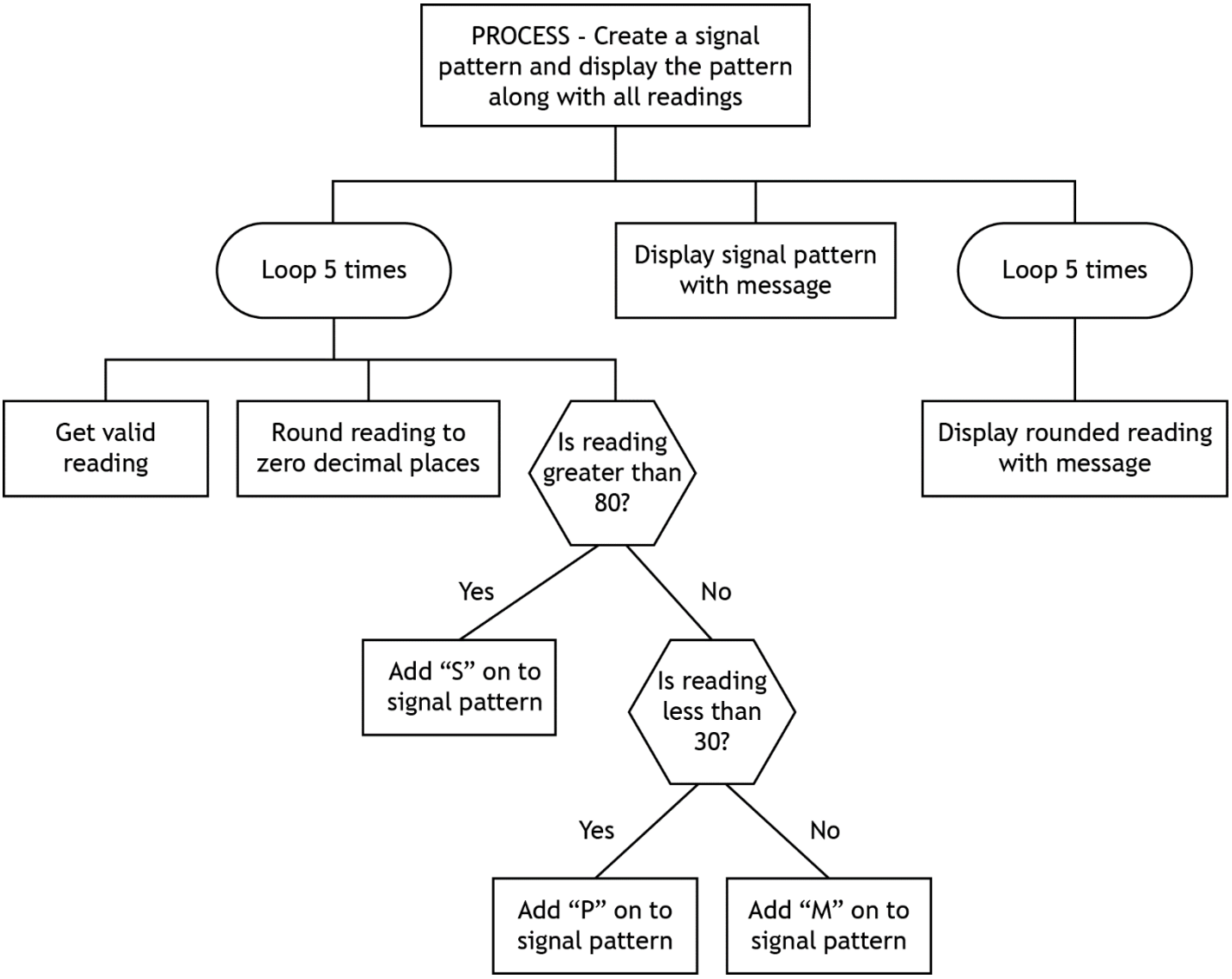
Processes

- round each reading to zero decimal places
- create a five character string representing the signal pattern

Outputs

- a message displaying the signal pattern
  for example - Signal Pattern is: SSMPS
- the five rounded readings with each reading number
  for example - Reading 1 - 89
  Reading 2 - 82
  Reading 3 - 56
  Reading 4 - 12
  Reading 5 - 99

Program design (structure diagram)



Task 2: software design and development

2a Using the program analysis and design, implement the program in a language of your choice. Ensure the program matches the structure diagram provided.

(15 marks)

Print evidence of your program code.

```
#2018 Assignment Solution

reading=[0.0]*5
signal = ""
index =0

for counter in range(5): #Start loop for 5 times

#get valid reading between 0 and 100
  reading[counter] = float(input("Please enter valid reading"))
  while reading[counter] < 0 or reading[counter] > 100:
    reading[counter] = float(input("Please re-enter between 0 and 100"))

#round to 0 decimal places
  reading[counter] = round(reading[counter])

  #Create signal pattern - running total but with letters to create signal pattern
  #S strong above 80, M medium 30 - 79, P poor less than 30
  if reading[counter] > 80:
    signal = signal + "S"
  elif reading[counter]<30:
    signal = signal + "P"
  else:
    signal = signal + "M"

print("The signal pattern is " + signal)

for counter in range(5):
  print("Reading " +str(counter+1)+ " : " + str(reading[counter]))
```

2b Your program should be tested to ensure it produces different signal patterns correctly.

Complete the table below to create one set of test data that will produce the expected output for the signal pattern shown.

(2 marks)

Type of test	User Input		Expected output for signal pattern	Actual output
Normal	reading 1	50	Signal pattern is: MPSPS	Please enter valid reading 50 Please enter valid reading 23 Please enter valid reading 87 Please enter valid reading 14 Please enter valid reading 90 The signal pattern is MPSPS Reading 1 : 50 Reading 2 : 23 Reading 3 : 87 Reading 4 : 14 Reading 5 : 90
	reading 2	23		
	reading 3	87		
	reading 4	14		
	reading 5	90		

You must demonstrate that your program correctly outputs the signal pattern and the rounded readings.

Print evidence of inputs and outputs to show that you have completed the test.

2c Your program should be tested to ensure that each signal strength character is correctly assigned as S, M or P. Six extreme test values are required to test this fully.

State the six test data values required:

(3 marks)

Extreme 1 0
Extreme 2 30
Extreme 3 80
Extreme 4 100
Extreme 5 81
Extreme 6 29

Candidate name_____ Candidate number_____

2d With reference to your code, evaluate your program by commenting on the following:

Fitness for purpose **(1 mark)**

I feel my program is fit for purpose as it successfully executes all the requirements of the task. My program also matches the supplied design.

My solution:

- allows the user to input the 5 readings.
- Validates the readings between 0 and 100
- Determine whether the reading is S (Strong) M(Medium) and P(Poor)
- Displays the signal reading correctly as demonstrated by my testing
- Displays all other relevant information to the screen

Where your code demonstrates efficient use of programming constructs **(1 mark)**

Efficiency answers may refer to:

- loops used instead of five individual inputs or outputs
- single variable only required for signal pattern rather than array of characters
- complex selection structure could have been used in place of separate "ifs"
- array used instead of five variables

Robustness of your completed program **(1 mark)**

My code is robust as it can cope with normal, extreme and exceptional test data. I have tested it thoroughly using a variety of test data, as demonstrated in my screenshots.

It can cope with negative numbers and invalid input and allows the user to re-enter a valid number if their input is out with the range 0 to 100

The only thing that crashes my program is when you type in a letter when the program expects a number. This is out with the requirements for this task.

Readability of your code **(2 marks)**

All evaluations must contain an element of evaluation rather than simple statements of terms. For example "I have used white space to highlight structures in my program" not "I have used white space".
Sample answer

I have used meaningful variables within the context of this task. This will improve the maintainability of the program and is easier to identify specific variables.

I have Internal commentary throughout my program that explains the main constructs and functions of my code.

I have used indentation so it easier to see where my constructs start and end.

Candidate name_____ Candidate number_____