



COMPUTING

# INDUCTION ASSIGNMENT

# INDUCTION ASSIGNMENT FOR COMPUTING

## Assignment

---

Students are expected to produce work with a minimum font size of 11. A bibliography should also be included at the end, to reference any sources used.

It is expected that your work will have your name in the header and that each Task is clearly labelled. Screenshots can be printed and enlarged as appropriate, provided that no required information is removed. There are up to 2 marks awarded for adhering to Standard Ways of Working.

### **TASK 1** (6 marks)

Explain, using examples, the meaning of programs, algorithms and source code.

### **TASK 2** (4 marks)

Discuss why are there so many different programming languages?

### **TASK 3** (4 marks)

What are the difference between low level and high level programming languages?

### **TASK 4** (2 marks)

What is indentation and why is it important when writing programs?

### **TASK 5** (12 marks)

Produce a simple diagram to explain the differences between an interpreter and a compiler

---

*For the remaining tasks, use a suitable programming language, ie Python. You will also need to screenshot both the source code and the results of the program when run, using different test data to show different outcomes when appropriate.*

### **TASK 6** (10 marks)

Write a program to ask your name and also how you are feeling. The output should then look like:

Hello *Steve*, so you are feeling *great* today.

### **TASK 7** (10 marks)

Write a program to ask the user for a starting even number and an ending number, so the computer can tell the user all the even numbers in between. Hint: You may need to use the *int* function

# INDUCTION ASSIGNMENT FOR COMPUTING

## TASK 8 (10 marks)

Write a program to ask the user to guess a number from 0 to 9. If the user guesses a 7, they are told they are correct. Otherwise they are asked to guess again until the correct answer ie 7 is given.

HINT: Use *while* command.

## TASK 9 (10 marks)

Write a program to ask the user to guess a number from 0 to 9. If the user guesses a 7, they are told they are correct. Otherwise they are told if their guess is too low or too high.

*Remember to screenshot both the source code and the result of the program when run, using different test data to show the different outcomes.*

## TASK 10 (15 marks)

Correct the following portion of code – indicate in red the changes and/or annotate changes

```
number = 17
run = 17
while run
guess = int(input(Enter a number:))
if guess = number:
print 'Yes.'
run = False
elseif guess > number
```

## TASK 11 (15 marks)

This will be a timed practical assessment during lesson time. You will be asked to write a program of your choice which demonstrates some of the skills you have learnt from this assignment. Your program can be simple but should take some form of user input and provide appropriate output(s). There will also be marks for the overall user experience. Your code should include brief explanatory comments as appropriate.

There will be no access to the Internet during this assessment and you will not be permitted to bring in any notes or code/programs already written.

Submission of this task will take place in lesson by email/memory stick which the teacher can then use to test the program. No annotation or screenshots are required.

# INDUCTION ASSIGNMENT FOR COMPUTING

## Assessment

---

The assignment will be assessed by your teacher and graded as High Quality, Good, Satisfactory or Unsatisfactory. The grade will be used alongside other assessments to decide whether students are on the appropriate course early on. You will also receive a final overall mark.

## Completion dates

---

The completion date for this assignment is the **Fri 12 September**. It should be submitted by hard copy to your teacher. Contact details of teaching staff will be given to students at the start of the course in September.

## Further Study/Websites (including online books)

---

[www.thestudentroom.co.uk](http://www.thestudentroom.co.uk)

<http://www.bbc.co.uk/news/technology/>

<http://www.eweek.com/c/a/IT-Management/10-Programming-Languages-You-Should-Learn-Right-Now/>

<http://www.sitepoint.com/best-programming-language-of-2013/>

<http://www.cs.waikato.ac.nz/~marku/languages.html>

<https://www.python.org/>

<http://greenteapress.com/thinkpython/thinkpython.html>

<http://thepythongamebook.com/en:pygame:start>

<http://www.itmaybeahack.com/homepage/books/python/html/index.html>

<http://inventwithpython.com/>