



---

# COMPUTER SCIENCES: PROGRAMMING

---

Worksheet for Week 1: Diving Right In! / Program Flow Control



JULY 6, 2020

TEESSIDE UNIVERSITY  
Borough Road, Middlesbrough, TS1 4PA

## Computer Sciences: Programming - Diving Right In!

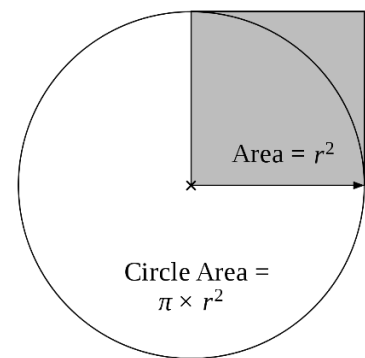
**Watch the week 1 part 1 video lecture (when available) before attempting these exercises.** This session aims to familiarise you with the IDLE environment and enable you to write simple Python modules using the `input()` and `print()` functions. You'll also work with mathematical operators and cast between data types.

**Hint:** Python comes with the value of the mathematical constant pi ( $\pi$ ) built in as `math.pi`. You'll be making heavy use of this today. Remember, you need to add `import math` at the top of your file to be able to use this!

### Question 1: Area of a Circle

You're tasked with writing a program to calculate and output the area of a circle.

1. Launch IDLE and create a new module file called "circlearea.py". See this week's lecture material if you're not sure how to do this.
2. Write Python code in your module to ask the user for the radius of a circle. The program will then calculate, store and output the area of the circle to the user. You can find the formula for the area of a circle here under the heading "Area": <https://www.mathsisfun.com/geometry/circle.html>
3. Modify the program to also store and output the circumference (perimeter) of the circle. You can find the formula at the same link under the title "Radius, Diameter and Circumference".



### Question 2: Area of Ellipse

You're tasked with writing a program to calculate and output the area of an ellipse (oval).

1. Start with your solution to question 1. Copy it to a new file called "ellipsearea.py".
2. Modify the program to take a second radius for an ellipse. The program should then calculate, store and output the area of the ellipse to the user. You can find the formula for the area of an ellipse here under the heading "Area": <https://www.mathsisfun.com/geometry/ellipse.html>

### Question 3: Volume of Cylinder

You're tasked with writing a program to calculate and output the volume of a cylinder.

1. Create a fresh module called "cylindervolume.py".

2. Write a program to compute the volume of a cylinder, asking the user for the radius and height. You can find the formula for the volume of a cylinder here: <https://www.mathsisfun.com/geometry/cylinder.html#Volume>

#### Question 4: Square Footage of a House

You're tasked with writing a program to calculate and output the square footage of a house.

1. Write a program that asks the user for the width and length of 4 rooms of a house, you can assume that the rooms are rectangular (i.e. no alcoves, etc). Once all the information has been collected, the program will then display the square footage per room and then the total square footage for the entire house.
2. Copy your solution and modify it to also ask the height of each room so that it can calculate the volume for each room, and the entire house.
3. Use the usual `width * length` formula for calculating the area of a rectangle and `width * length * height` for calculating the volume of a cuboid.

#### Question 5: Fahrenheit-Celsius Conversion

Write a program that prompts the user to input a temperature in Fahrenheit. The program will then output the temperature in Celsius. The format of the output is required to be:

```
>> 71 Degrees Fahrenheit is 21.67 degrees Celsius
```

Get the formula for the conversion here:

<http://www.calculateme.com/cTemperature/FahrenheitToCelsius.htm>

#### Question 6: Celsius-Fahrenheit Conversion

Write a program that prompts the user to input a temperature in Celsius. The program will then output the temperature in Fahrenheit. The format of the output is required to be:

```
>> 22 Celsius is 71.6 degrees Fahrenheit.
```

Get the formula for the conversion here:

<http://www.calculateme.com/cTemperature/CelsiusToFahrenheit.htm>

## Advanced Activity: Days Elapsed

You are required to write a Python program which implements the following instructions for calculating the days elapsed between two dates. The calculations are based on a simple estimation algorithm that avoids using look-up tables for leap years and days in months.

### Steps

1. Begin your Python program (module) with a comment to explain what the program does. It's good professional practice to document your code thoroughly.
2. Include your name and today's date.
3. Assign the following values to the given variables:  
`xmas_day = 25`  
`xmas_month = 12`  
`xmas_year = 2020`
4. Now write three `input()` statements to acquire information about your birthday. These input statements should clearly ask for your day, month and year of birth and assign the values entered to the following variables: `born_day`, `born_month`, `born_year`. The month should be the number of the month, with January being 1 and December being 12.
5. Now write three more input statements to acquire today's date and assign the responses to the following variables: `now_day`, `now_month` and `now_year`.
6. Now calculate the number of days elapsed since your birthday. You will need three new variables: `years_indays`, `months_indays`, `days_born`.
7. First, convert years to days using the following formula:  
`years_indays = (now_year - born_year) * 365.25`
8. Now calculate an adjustment for months using the following formula:  
`months_indays = (now_month - born_month) * 30.4`  
`+ now_day - born_day`
9. Finally calculate elapsed days (`days_born`) as:  
`days_born = years_indays + months_indays`
10. Repeat the calculations for days to Christmas this year, assigning the number of days to Christmas to a variable called `days_toxmas`. You will need to substitute the appropriate variables in the formulae given above to re-calculate `years_indays` and `months_indays` to estimate days to Christmas:  
`years_indays = (xmas_year - now_year) * 365.25`  
`months_indays = (xmas_month - now_month) * 30.4 + xmas_day`  
`- now_day`  
`days_toxmas = years_indays + months_indays`

11. Finally, use a `print()` statement to output the results with the following messages:

```
Approximately days_born days have elapsed since your birthday!  
And it is only days_toxmas days to Xmas!
```

Be careful with your data types and make appropriate use of casting!

## Computer Sciences: Programming - Program Flow Control

**Watch the week 1 part 2 video lecture (when available) before attempting these exercises.** This session aims to familiarise you flow control, lists and iteration.

**Hint:** It's good practice to plan your programs first on paper using *pseudocode*. When we say pseudocode, we mean code that is halfway between English and a programming language, such as Python. Read more about it here:  
<https://en.wikipedia.org/wiki/Pseudocode>

### Question 1: Comparing Numbers

You're tasked with writing a program to compare two numbers.

1. On paper, devise a routine using pseudocode to ask the user for two numbers, compare them and then print a message indicating which one is bigger than the other (or if they are both the same) with an appropriate message.
2. Carefully work out the structure of your conditional (`if`) statement before you begin implementation of your procedure.
3. Now implement your algorithm in Python!

### Question 2: Eligible Voters

You're tasked with writing a program to check if someone is eligible to vote.

3. Create a program to ask the user for their name, age and country of citizenship. Your program should then process this information to determine if the named person can vote according to the following rules:
  - a. If the person is under 18, they are not allowed to vote. Print out a message which includes their name and states that they are not yet old enough to vote.
  - b. If the person is over 18 and their country of citizenship is the UK or Britain, print out a message including the name that states they are entitled to vote.
  - c. Otherwise, if they are over 18 but their country of citizenship is not the UK or Britain, print out a message stating that they are not entitled to vote.
4. Test your program with suitable test cases.

### Question 3: Fizz Buzz

You're tasked with creating a program that asks the user for number and then outputs a message based on the properties of that number:

3. Ask the user for a number and store it in a variable.
4. If the number is divisible by 3, print the message `Fizz`.
5. If the number is divisible by 5, print the message `Buzz`.

6. If the number is divisible by both 3 and 5, print the message `Fizz Buzz`.
7. If the number is not divisible by either 3 or 5, print the number itself.

**Hint:** Think about how you will test if a number is divisible by another number. Do some research on the modulo operator (%) and think carefully about how it can help us here. Plan your conditional statements out on paper before you get started coding.

#### Question 4: Sorting

You're tasked with writing a program that sorts 3 numbers.

1. Create a program to ask the user for three numbers (called `first_num`, `second_num`, `third_num`) and print them out in order from smallest to largest.
2. Test your code using the following test cases. You must enter the numbers in the exact order given for each test case. So, for the first test case (test case a), `first_num = 87`, `second_num = 99`, and `third_num = 11`. Run your program for each test case to ensure it works correctly.
  - a. 87, 99, 11
  - b. 87, 11, 99
  - c. 99, 87, 11
  - d. 99, 11, 87
  - e. 11, 99, 87
  - f. 11, 87, 99

#### Question 5: Grades

You're tasked with writing a program that assigns grades to students based on their marks:

1. Create a program that asks the user for a mark from 0 to 100 and prints a corresponding grade using the table to the left.
2. Devise a test case scheme and test that your program produces the correct output!
3. This might be trickier than you think. What if the user enters a negative number? What if they enter a number greater than 100? What if the thing they enter isn't a number at all?

Mark	Grade
0	N/S
1-39	F
40-49	D
50-59	C
60-69	B
70-79	A
80-100	A*

### Question 6: Who's that Pokémon?

You're tasked with writing a program that asks the user for the type of a Pokémon and the first letter of its name, then makes suggestions about what that Pokémon might be called.

1. Ask the user for the type of the Pokémon. This can be one of:
  - a. Fire
  - b. Water
  - c. Grass
  - d. Electric
2. Ask the user for the first letter of that Pokémon's name.
3. Based on the information you've got, output suggestions based on the following table.

Type	First letter	Suggestion
Water	S	Squirtle
Water	T	Tentacool
Fire	C	Charmander
Fire	M	Moltres
Grass	B	Bulbasaur
Grass	O	Oddish
Electric	P	Pikachu
Electric	V	Voltorb

### Question 7: Comparing Many Numbers

You're tasked with improving on your answer to question 1 by allowing multiple comparisons per program run.

1. Copy your answer to question 1 and save it in a different file.
2. Adjust your answer to use a loop. Keep asking for pairs of numbers and comparing them until the user types "quit" instead of a number.
3. Pay particular attention to your casts here. Can you wrap the casts around your `input()` calls and still have the program work as expected? If not, where should you put them instead?

### Question 8: Sum of Numbers at Even Indices

You're tasked with summing up only the numbers in a list that are at even indices.

1. Start a fresh file.
2. Declare a list of integers containing the values 14, 5, 19, 20, 21, 66 and 89 in that order.
3. Using a loop, add up only the values at even indices in this list. So that's:
  - a. 14 (index 0)
  - b. 19 (index 2)
  - c. 21 (index 4)



- d. 89 (index 6)
- 4. Don't forget to make use of the modulo operator for determining number evenness. Use a `while` loop, consult this week's lecture for a hint (slide 19).

### Advanced Activity: Bank Consultancy

You're tasked with writing a program to make decisions on loan applications.

1. Your program must ask the user for 4 values:
  - a. Credit score (0 to 10 inclusive)
  - b. Address term (months at present address)
  - c. Income (£s)
  - d. Request (£s) (loan amount requested)
2. Your program must then process this data, then print out the input data and a message stating whether the loan request has been granted or not.
3. In conversation with the bank manager, you were given the rules below.
4. You will need to think about organising these and devising an algorithm before you begin coding.

#### The Rules

If income or credit score is zero, or time at present address is less than 12 months, the request is refused.

If the requested loan amount is greater than the applicant's income, the loan is only granted if the request is less than twice the income and the applicant has lived at their present address for 60 months or more and the credit score is 5 or greater.

If the credit score is 7 – 10 inclusive, and time at present address is greater than 12 months but less than 60, and the request is less than the applicant's income, grant the loan.

If the credit score is between 2 and 5 inclusive and the applicant has been at their present address for 60 months or more and the requested loan is less than the applicant's income, grant the loan.

If the credit score is 1 and the applicant has been at their present address more than 12 months and the requested loan is less than 20% of the applicant's income, grant the loan.

In all other cases, the application is refused.