Python Programming

# Exercise 1

1. **Type this into PyScripter and run it.**
2. **Write a comment above each line explaining what it does. Remember that = does not mean equals!**
3. **In your comments state the data type of each variable (Floating, Integer, String or Boolean).**

cars = 100

space\_in\_a\_car = 4.0

drivers = 30

passengers = 90

cars\_not\_driven = cars - drivers

cars\_driven = drivers

carpool\_capacity = cars\_driven \* space\_in\_a\_car

average\_passengers\_per\_car = passengers / cars\_driven

print ("There are ", cars, "cars available.")

print ("There are only", drivers, "drivers available.")

print ("There will be", cars\_not\_driven, "empty cars today.")

print ("We can transport", carpool\_capacity, "people today.")

print ("We have", passengers, "to carpool today.")

print ("We need to put about", average\_passengers\_per\_car, "in each car.")

**Here’s what you should see:**

*There are 100 cars available.*

*There are only 30 drivers available.*

*There will be 70 empty cars today.*

*We can transport 120.0 people today.*

*We have 90 to carpool today.*

*We need to put about 3.0 in each car.*

1. **Change the first print line to this:** print ("There are \t", cars, "cars available.")
**What does \t do?
Try \n and then \r.**

# Exercise 2

1. **Type this into PyScripter and run it. Test the program with 212o, 100o, 0o, 7o,  -28o . You can only truly test it if you know what the expected outcomes are!**
2. **Can you suggest some additional test values that would test the program more thoroughly?**
3. **Write a comment above each line explaining what it does.**

|  |  |  |
| --- | --- | --- |
| Temperature Input | oF | oC |
| 212 |  |  |
| 100 |  |  |
| 0 |  |  |
| 7 |  |  |
| -28 |  |  |
|  |  |  |
|  |  |  |

temp = float(input("Give me a temperature : "))

degrees\_c = (temp - 32) \* 5 / 9

degrees\_f = temp \* 9 / 5 +32

print ( temp, "degrees F = ", degrees\_c , " degrees C")

print ( temp, "degrees C = ", degrees\_f , " degrees F")

**Here’s what you should see:**

*212.0 degrees F = 100.0 degrees C*

1. *degrees C = 413.6 degrees F*
2. **Change the first print line to the one below and see what difference it makes.**
3. **Similarly, change the other print line.**

temp = float(input("Give me a temperature : "))

degrees\_c = (temp - 32) \* 5 / 9

degrees\_f = temp \* 9 / 5 +32

print ( temp, "degrees F = ", "%.2f" % degrees\_c , " degrees C")

print ( temp, "degrees C = ", degrees\_f , " degrees F")

# Exercise 3

1. **Type this into PyScripter and run it. Test the program with 212o, 100o, 0o, 7o,  -28o . You can only truly test it if you know what the expected outcomes are!**
2. **Can you suggest some additional test values that would test the program more thoroughly?**
3. **Write a comment above each line explaining what it does.**

answer = "Y"

while answer == "Y":

 temp = float(input("Give me a temperature : "))

 degrees\_c = (temp - 32) \* 5 / 9

 degrees\_f = temp \* 9 / 5 +32

 print ( temp, "degrees F = ", "%.2f" % degrees\_c , " degrees C")

 print ( temp, "degrees C = ", "%.2f" % degrees\_f , " degrees F")

 answer = input ("Do you want to run this again (Y or N)? ")

print ("Thank you and goodnight")

**Here’s what you should see:**

212.0 degrees F = 100.00 degrees C

212.0 degrees C = 413.60 degrees F

-7.0 degrees F = -21.67 degrees C

-7.0 degrees C = 19.40 degrees F

Thank you and goodnight

1. **Change line 2 to this, investigate the difference it makes and update your comment line.**

 while answer == "Y" or answer == "y" :

## Looping StructureWhile



## Summary of Operators

| **Operation** | **Meaning** |
| --- | --- |
| < | strictly less than |
| <= | less than or equal |
| > | strictly greater than |
| >= | greater than or equal |
| == | equal |
| != | not equal |
| Is | object identity |
| is not | negated object identity |

Arithmetic Operators

| **Operation** | **Result** |
| --- | --- |
| x + y | sum of *x* and *y* |
| x – y | difference of *x* and *y* |
| x \* y | product of *x* and *y* |
| x / y | quotient of *x* and *y* |
| x // y | floored quotient of *x* and *y* |
| x % y | remainder of x / y |
| -x | *x* negated |
| +x | *x* unchanged |
| abs(x) | absolute value or magnitude of *x* |
| int(x) | *x* converted to integer |
| float(x) | *x* converted to floating point |
| complex(re, im) | a complex number with real part *re*, imaginary part *im*.*im* defaults to zero. |
| c.conjugate() | conjugate of the complex number *c* |
| divmod(x, y) | the pair (x // y, x % y) |
| pow(x, y) | *x* to the power *y* |
| x \*\* y | *x* to the power *y* |

Looping Structure

## For



## Exercise 4

1. **Type this into PyScripter and run it. Test the program with positive numbers**Note that to use Maths functions that are not listed in the table above, you have to IMPORT a MATHS module. See line 1
2. **Write a comment above each line explaining what it does.**

import math

n = float(input("Give me a positive number : "))

print ("Number\tSq Root")

print (n,"\t\t%.4f" % math.sqrt(n))

1. **Type this into PyScripter and run it.**
	1. NB import math has moved – does it still work? Try moving it to the end – does it still work?
	2. Also note the subtle change to the print line. What difference does it make?
2. **Type this into PyScripter and run it.**
3. **Write a comment above each line explaining what it does.**

 import math

 print ("Number\tSq Root")

 print ()

 for n in range (1 , 10 ):

 print (n, "\t\t" "%.4f" % math.sqrt(n))

1. **Experiment with different numbers in line 4.**

## Exercise 5

1. **Type this into PyScripter and run it. Test the program fully with suitable test data. You can only truly test it if you know what the expected outcomes are!**
2. **Write a comment above each line explaining what it does.**

print(" Discount Calculator. There is 12.5% discount on items over £60")

cost=float(input("Ticket price"))

if cost > 60 :

 discount = cost \* .125

else :

 discount = 0

print ("Your discount is £%.2f" % discount)

print ("The price you pay is £%.2f" % (cost - discount))

## Exercise 6

1. **Write a program to produce a conversion table from inches to centimetres (1” is 2.54 cm ). The table should list inches from 1 to 12 in steps of 1.**
2. **Adjust the above program to one where the user inputs the start number of inches and the end value.**
3. **Adjust the program in 2 to go from 1 to 12 in steps of 0.1”**

## Exercise 7

**Write a program to grade percentage test scores which the user enters during execution. To stop the program, the user will enter a negative score. The grades boundaries are:**

|  |  |
| --- | --- |
| Grade | Mark |
| Distinction  | >90 |
| Merit  | >80 |
| Pass | >45 |
| Fail | <=45 |

## Exercise 8

Write a guessing game program. The ‘computer’ thinks of a number between 1 and 100 inclusive. You type in your guess and the ‘computer’ will tell you if your guess is correct (and the program then ends) or your guess is too high or too low (and the program give you the appropriate feedback).

At the end of the program it will tell you how many guesses it took you to get the correct answer.