



# Introduction to Programming

---

Department of Computer Science and Information  
Systems

Lecturer: Steve Maybank  
[sjmaybank@dcs.bbk.ac.uk](mailto:sjmaybank@dcs.bbk.ac.uk)  
Spring 2020

Revision of the Summer 2017 Examination



# Question 1a

---

Which of the following are names of variables

\_3y6

3y\_6

Y3-6

\_ \_ # adjacent underscores

p08\_



# Rules for the Names of Variables

---

- The only characters allowed are numbers, letters and underscore
- A name cannot begin with a number



# Question 1b

---

- What are the final values of a, b, c, d, e?
- $p = 3$
- $q = 2$
- $a = p + q * 4$
- $b = 10 ** p ** q$
- $c = 5 * p - q / 5$
- $d = 5 - (4 - (3 - (2 - p)))$
- $e = p \% 2 + q \% 2$



# Operators and Precedence

---

- Exponentiation: \*\*
- Times: \*      Real Division: /      Remainder: %
- Plus: +      Minus: -
  
- Apply the highest precedence operator first:
  - $25\%2^{**}3$
  - If the operators have the same precedence then evaluate left to right:
    - $25\%3^*8$



## Question 2b

---

- A swimming pool with volume `poolVolume` litres is initially empty. Water enters at `inSpeed` litres/min, but leaks away at `outSpeed` litres/min. Find the number `mFill` of minutes required to fill the pool.
- $(\text{inSpeed} - \text{outSpeed}) * \text{Mfill} = \text{poolVolume}$
- Note the camel case names



# Volume of Water

---

- The volume entering the pool in one minute is  $(\text{inSpeed} - \text{outSpeed}) * 1$
- The volume entering the pool in mFill minutes is  $(\text{inSpeed} - \text{outSpeed}) * \text{mFill}$
- The definition of mFill ensures that the last quantity is the pool volume.



## Question 2c

---

- A train with an average speed of 100km per hour travels from L to B in 3 hours, but it has to wait at B. As a result, the average speed is reduced to 90km per hour. Find the waiting time.
- $90 = \text{distance}/\text{totalTime}$   
 $= \text{distance}/(\text{journeyTime} + \text{waitingTime})$   
 $= 300/(3 + \text{waitingTime})$
- $\text{waitingTime} = 1/3$  hours





# Question 3

---

- Find five errors in the code

```
import = 0          # import is a reserved word
if export > import : # export not defined
    import = 3.0
    export = 3E-5
    print("import is"+3.0) # string+number
else                # missing colon
    export = export//import # division by zero
```



# Question 4

---

- Write code to show "Please enter the price:" in the shell and to return a value entered at the keyboard. The returned value is converted to floating point and assigned to a variable price.
- `price = float(input("Please enter the price:"))`



# Question 4, Second Part

---

- What is the value of price for each of these inputs in the shell

Please enter the price: 5	# 5.0
Please enter the price: "5"	# error
Please enter the price: 6.2	#6.2
Please enter the price: 5*6.2	# error



## Question 4, Part c (i)

---

- What is printed when the following code is run?

```
print("He said\n\"Hello\" and 'Goodbye'\ntoday.")
```

```
"He ... today." # The quotes contain a string  
said\n          # \n is new line  
\"Hello\"    # \" is a character in a string  
'Goodbye'   # ' is a character in a string  
\ntoday      # \n is new line
```



## Question 4, Part c (ii) and (iii)

---

- What is printed when the following code is run?

```
houseNumber = 32
```

```
streetName = "Hamilton Road"
```

```
print(houseNumber + streetName)
```

```
print(len("\\\\"*3+"Harry"))
```

```
"\\"      # string with the single character backslash
```

```
"\\"*3    # repeat the string three times
```

```
+        # string concatenation
```



## Question 4, Part c (iv) and (v)

---

- What is printed when the following code is run?

```
firstName = "Mickey"  
lastName = "Mouse"  
print(firstName[3], lastName[-3])
```

```
names = ["Amir", "Barry", "Chris", "Dylan"]  
print(names[-1][-1])
```



## Question 5a

---

- What is printed? Use ~ to indicate a space

```
temperature = 39.8954
```

```
print("A:", "%d" % temperature)
```

```
A:~39 # %d implies integers
```

```
print("B:", "%.f" % temperature)
```

```
B:~40 # %.f implies floating point, no fraction
```



## Question 5a Continued 1

---

- What is printed? Use ~ to indicate a space

```
temperature = 39.8954
```

```
print("C:", "%s" % temperature)
```

```
C:~39.8954 # %s implies string
```

```
print("D:", "%06.2f" % temperature)
```

```
D:~039.90 # %06.2f implies a field width
```

```
# of 6, 2 decimal places and 0 padding
```





## Question 5a Continued 2

---

- What is printed? Use ~ to indicate a space

```
temperature = 39.8954
```

```
print("E:", "%-7d%%" % temperature)
```

```
E:~39~::~~% # %-7d implies integer
```

```
# in a field of width 7 and left justified.
```

```
# %% implies a single character %
```



## Question 5b

---

- Write code to print the following using a multiple format specifier

Temperature:~39.9~C~~Feeling:~~~Great

```
st1 = "Temperature:~%4.1f~C~~Feeling:~%7s"
```

# %4.1f implies a field of width 4 and 1 decimal place

# %7s implies a string in a field of width 7

```
print(st1 % (temperature, "Great"))
```



# Question 6a

---

- Explain how the lexicographic ordering for characters is extended to give a lexicographic ordering for strings
- Let  $a$ ,  $c_1$ ,  $c_2$  be any strings. Let  $b_1$ ,  $b_2$  be distinct characters, i.e. strings of length 1.  
$$st_1 = a + b_1 + c_1$$
$$st_2 = a + b_2 + c_2$$
- If  $b_1 < b_2$  then  $st_1 < st_2$ . If  $b_1 > b_2$  then  $st_1 > st_2$



# Question 6b

---

- space < numbers < uppercase < lowercase
- True or False:

"1" < "2"

"cart" < "carp"

"milestone" < "mile stone"

"Corn" < "2Corn"

"car" < "care"



# Question 7a

---

wind speed	description
$s \leq 1$	calm
$1 < s \leq 3$	light air
$3 < s \leq 7$	light breeze
$7 < s$	out of range

- Write nested if statements to print the correct descriptions given a non-negative value of  $s$ . Use `if ... elif ... else`.



# Question 7a Continued

---

wind speed	description
$s \leq 1$	calm
$1 < s \leq 3$	light air
$3 < s \leq 7$	light breeze
$7 < s$	out of range

- `if s <= 1:`
- `print("calm")`
- `elif s <= 3:`
- `print("light air")`
- `elif s <= 7:`
- `print("light breeze")`
- `else:`
- `print("out of range")`



# Question 8a

---

- What is printed when the following code is run?

```
sumSquares = 0
```

```
for x in range(3):
```

```
    sumSquares = sumSquares+x*x
```

```
print(sumSquares)
```

# x in range(3), so x takes the values 0, 1, 2



# Question 8b

---

- What is printed when the following code is run?

```
sumSquares = 0
```

```
for x in range(3):
```

```
    sumSquares = sumSquares+x*x
```

```
    print(sumSquares)
```

```
# x in range(3), so x takes the values 0, 1, 2
```





# Question 8c

---

Write code to input 6 numbers and then print out the position of the largest number.

```
large = float(input("Enter a number:"))
position = 1
for i in range(2, 7):
    x = float(input("Enter a number:"))
    if x > large:
        large = x
        position = i
print("position of largest no:", position)
```



# Question 9a

---

- Identify in the following code the name of a variable, the name of a function and the values of the arguments to a function

```
quantity = 5*8
```

```
price = round(6.8275, 2)
```



# Question 9c

---

describe the  
compile time  
error in the  
code.

```
def main():  
    result = cubeVolume(2)  
    print("side length 2, volume", result)
```

```
main()
```

```
def cubeVolume(sideLength):  
    if (sideLength <= 0) :  
        return 0  
    return sideLength**3
```



# Question 9d

---

- Identify the branches in the function cubeVolume

```
def cubeVolume(sideLength):  
    if (sideLength <= 0) :  
        return 0  
    volume = sideLength**3  
    return volume
```



# Question 10b

---

- What is the length of the list scores? Which non-negative integers can be indices to scores?

scores = [17, 5, 3]



# Question 10c

---

- Which number is printed by the following code?

```
scores = [17, 5, 3]
```

```
values = scores
```

```
scores[0] = 10
```

```
print(values[0])# two names but only one list of numbers
```