



Grade 6 Math Circles

Computer Science - Solutions

Exercise Solutions

Activity 1

What are the values of a , b , c and d after the following code is run?

```
a = 0
b = a
d = b
a = 7
c = a
```

Activity 1 Solution

We will keep track of each value after each step using the table below:

	Initial	Line 1	Line 2	Line 3	Line 4	Line 5
a	None	0	0	0	7	7
b	None	None	0	0	0	0
c	None	None	None	None	None	7
d	None	None	None	0	0	0

Thus, after the code is run, we have that $a = 7$, $b = 0$, $c = 7$ and $d = 0$.

Activity 2

Determine the data type of the following values in Python.

- (a) -4839.1
- (b) "76"
- (c) 0
- (d) 'False'



Activity 2 Solution

- (a) **float**, since numerical value has a decimal
- (b) **str**, since value is enclosed in quotations
- (c) **int**, since numerical value has no decimal
- (d) **str**, since value is enclosed in quotations

Activity 3

In order to convert temperature from Celsius to Fahrenheit, first multiply the temperature in Celsius by 1.8 and then add 32. Write a Python program called *celsius_to_fahrenheit* that inputs a temperature in Celsius and outputs the corresponding temperature in Fahrenheit. What is the corresponding Fahrenheit temperature for the following?

- (a) 0 °C
- (b) 100 °C
- (c) 18 °C
- (d) -40 °C

Activity 3 Solution

```
def celsius_to_fahrenheit(c_temp):  
    f_temp = (c_temp * 1.8) + 32  
    return f_temp  
  
celsius_to_fahrenheit(0)  
celsius_to_fahrenheit(100)  
celsius_to_fahrenheit(18)  
celsius_to_fahrenheit(-40)
```

- (a) 32 °F
- (b) 212 °F
- (c) 64.4 °F
- (d) -40 °F



Problem Set Solutions

1. Determine the values of each of the variables and their data types after the following code is run.

```
a = 17
b = 5.0
c = b - a
d = b * c
e = (a != d)
f = not(not(e))
g = e and not(f)
b = d
```

Solution: We will keep track of each value after each step using the table below:

	Initial	Line 1	Line 2	Line 3	Line 4	Line 5	Line 6	Line 7	Line 8
<i>a</i>	None	17	17	17	17	17	17	17	17
<i>b</i>	None	None	5.0	5.0	5.0	5.0	5.0	5.0	-60.0
<i>c</i>	None	None	None	-12.0	-12.0	-12.0	-12.0	-12.0	-12.0
<i>d</i>	None	None	None	None	-60.0	-60.0	-60.0	-60.0	-60.0
<i>e</i>	None	None	None	None	None	<i>True</i>	<i>True</i>	<i>True</i>	<i>True</i>
<i>f</i>	None	None	None	None	None	None	<i>True</i>	<i>True</i>	<i>True</i>
<i>g</i>	None	None	None	None	None	None	None	<i>False</i>	<i>False</i>

Thus, after the code is run, we have that $a = 17$ (**int**), $b = -60.0$ (**float**), $c = -12.0$ (**float**), $d = -60.0$ (**float**), $e = \text{True}$ (**bool**), $f = \text{True}$ (**bool**) and $g = \text{False}$ (**bool**).

2. Let $a = 25$, $b = 4$ and $c = 9$. Determine the following.

(a) $a + b - c$

(b) $a * (c / b)$

(c) $(b ** b) + (9 // 4)$

(d) $(a \% b) - (a * b * c)$



Solution:

$$(a) a + b - c \implies 25 + 4 - 9 \implies 29 - 9 \implies 20$$

$$(b) a * (c / b) \implies 25 * (9 / 4) \implies 25 * (2.25) \implies 56.25$$

$$(c) (b ** b) + (c // a) \implies (4 ** 4) + (9 // 4) \implies 256 + 2 \implies 258$$

$$(d) (a \% b) - (a * b * c) \implies (25 \% 4) - (25 * 4 * 9) \implies 1 - (100 * 9) \implies 1 - 900 \implies -899$$

3. Let $a = 3$, $b = -8$, $c = 3.0$ and $d = 0$. Determine the following.

(a) $\text{not}(a == c)$

(b) $(b <= c)$ and $(d > b)$

(c) $\text{not}((a != b) \text{ or } (d == 0))$

Solution:

$$(a) \text{not}(a == c) \implies \text{not}(3 == 3.0) \implies \text{not}(True) \implies False$$

$$(b) (b <= c) \text{ and } (d > b) \implies (-8 <= 3.0) \text{ and } (0 > -8) \implies True \text{ and } True \implies True$$

$$(c) \text{not}((a != b) \text{ or } (d == 0)) \implies \text{not}((3 != -8) \text{ or } (0 == 0)) \implies \text{not}(True \text{ or } True) \implies \text{not}(True) \implies False$$

4. The volume of a rectangular prism is determined by multiplying the *length*, *width* and *height* of the rectangular prism by each other. Write a program called `rec_prism_volume` that inputs the length, width and height of a rectangular prism, and outputs the volume of the rectangular prism.

Solution:

```
def rec_prism_volume(length, width, height):  
    volume = length * width * height  
    return volume
```

5. There are approximately 1.609344 kilometres in a mile. Write two programs:

- `km_to_miles`, inputs a distance in kilometres and outputs the equivalent distance in miles



- *miles_to_km*, inputs a distance in miles and outputs the equivalent distance in kilometres

Use these two programs to make the following conversions. Round to 4 decimal places.

- (a) 1 km to miles
- (b) 10 miles to km
- (c) 120 km to miles
- (d) 54.0592937 miles to km

Solution:

```
def km_to_miles(km_distance):  
    mile_distance = km_distance / 1.609344  
    return mile_distance  
  
def miles_to_km(mile_distance):  
    km_distance = mile_distance * 1.609344  
    return km_distance
```

- (a) 1 km = 0.6214 miles
- (b) 10 miles = 16.0934 km
- (c) 120 km = 74.5645 miles
- (d) 54.0592937 miles = 87 km

Bonus Questions

6. Suppose a person writes code that prints the sum of any two inputted numbers. Their code is shown below, but it contains an error. Determine if it is a syntax error or a semantic error, and specify where it occurs. How could this code be fixed? (Hint: Run it through Python Tutor).

```
num1 = input('Enter a number:')  
num2 = input('Enter another number:')  
  
sum = num1 + num2  
  
print('The sum of', num1, 'and', num2, 'is', sum)
```



Solution: This code contains a semantic error because it uses incorrect logic. The error occurs at the line

$$sum = num1 + num2$$

Since the values of *num1* and *num2* are user inputs, the data types of these values is **str**. So, when the third line of code runs, we have two strings being added together, which results in the value of *sum* being the value of *num1* followed by the value of *num2*. (e.g. If *num1* = “7” and *num2* = “6”, then *sum* = “76”).

We can fix this error by adding two lines of code, before *sum* is defined, where the values of *num1* and *num2* are converted to **int**. The correct code is shown below.

```
num1 = input('Enter a number:')
num2 = input('Enter another number:')

num1 = int(num1)
num2 = int(num2)

sum = num1 + num2

print('The sum of', num1, 'and', num2, 'is', sum)
```

7. The following code contains a total of 3 syntax errors. Identify each of them.

```
a == input("Enter a number:")
b = int(a)
c = a + b
a = int(a)
d = 0
e = a / d
```

Solution: Here are the syntax errors in the code:

- (1) The first line uses “==” and not “=”, which means that instead of defining the value of *a* to be the user inputted value, we are comparing the value of *a* to the user inputted value. Since *a* hasn’t been defined yet, we get a syntax error.
- (2) The third line adds the value of *a*, which is **str**, and the value of *b*, which is **int**, which causes a syntax error.
- (3) The sixth line divides the value of *a* by the value of *d*, which is 0, thus a division by 0 causes a syntax error.