PHY1112 Lab 2

Types and Operators

January 16th, 2024

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Part | 1 | 2 | 3 | 4 | 5 | 6 | Total |
| Points | 3 | 3 | 1 | 2 | 2 | 10 | 21 |
| Score |  |  |  |  |  |  |  |

Objectives

1. See examples of different types in Python, their behaviors, and how explicit and implicit casting between them works.
2. Create a quadratic equation calculator.

Part 1: Investigating types

1. (3 points) Create a Python file named “part1.py” and in it create the following variables using explicit typing:

|  |  |  |
| --- | --- | --- |
| Variable name | Variable type | Variable value |
| greeting | str | “Hello” |
| ultimate\_answer | int | 42 |
| something\_complex | complex | (24.5+17.5j) |

1. (3 points) In the same file, create the following variables using implicit typing.

|  |  |  |
| --- | --- | --- |
| Variable name | Variable type | Variable value |
| place | str | “World” |
| anti\_real\_ultimate\_answer | float | -42.0 |
| class\_info | dict | Key: “subject”, value: “PHY”  Key: “number”, value: 1112 |

1. (1 point) Create a new variable named real\_ultimate\_answer, which is your previous variable ultimate\_answer, but now cast as a float.
2. (2 points) After creating the variables for questions 1, 2 and 3, print out the values and types of all seven of your variables using type() and confirm that they are correct.

A screen shot of a computer program

Description automatically generated

1. (2 points) State what each of the following evaluate to:
   1. bool(0): False
   2. bool([]): False
   3. bool([0]): True
   4. bool(greeting): True

Part 2: Operators – creating a quadratic equation calculator

One of the oldest known solutions in mathematics is that which finds the roots of a second order polynomial, also known as a quadratic equation. While solutions likely date back to ancient Egypt, in modern terms we pose the problem as follows. Given a second order polynomial of the form

when , there exists either one or two roots of the form

.

Since calculating the roots of only requires elementary operations, we can write a program to do the calculation part for us.

1. (10 points) Create a new Python file called “part2.py” and write a quadratic equation solver in it.
   * + Define three variables at the start of your file, name them a, b, and c, and use the input() function to grab their values from the user. Note that the input function returns a string by default, and thus a, b, and c will need to be converted to floats.

Also make sure that you always give a nonzero value for a.

* + - For the square root, use \*\*0.5
    - The roots can be either real or complex. To better illustrate this, have your program print out the value for each of the roots, as well as the type for each of the roots.

Run your program and obtain the output (root value + types) for the following inputs:



A screenshot of a computer

Description automatically generated

**Code**

'''

Filename:       part1.py

Author:         Patrick Geraghty

Date Created:   2024-01-23

Date Modified:  2024-01-24

Description:    Contains variables with explicit and implicit typing, and type conversion.

'''

# Part 1

# Variables with explicit typing

greeting: *str* = "Hello"

ultimate\_answer: *int* = 42

something\_complex: *complex* = (24.5 + 17.5*j*)

# Variables with implicit typing

place = "World"

anti\_real\_ultimate\_answer = -42.0

class\_info = {"subject": "PHY", "number": 1112}

# Type conversion

real\_ultimate\_answer = *float*(ultimate\_answer)

# Print statements for type and value of each variable

print(*type*(greeting))

print(greeting)

print()

print(*type*(ultimate\_answer))

print(ultimate\_answer)

print()

print(*type*(something\_complex))

print(something\_complex)

print()

print(*type*(place))

print(place)

print()

print(*type*(anti\_real\_ultimate\_answer))

print(anti\_real\_ultimate\_answer)

print()

print(*type*(class\_info))

print(class\_info)

print()

print(*type*(real\_ultimate\_answer))

print(real\_ultimate\_answer)

'''

Filename:       part2.py

Author:         Patrick Geraghty

Date Created:   2024-01-24

Date Modified:  2024-01-24

Description:    Contains a program that takes the 'a', 'b', and 'c' values of a quadratic equation as parameters, and returns the roots of the equation as well as the type of roots (real and distinct, real and equal, or complex).

'''

# Input for variables 'a', 'b', and 'c'

a = *float*(input("Enter a: "))

b = *float*(input("Enter b: "))

c = *float*(input("Enter c: "))

# Print statements for the roots of the quadratic equation dependant on the discriminant

if (b\*\*2-4\*a\*c) < 0:

    print(*complex*((-b)/(2\*a), + (-(-(b\*\*2-4\*a\*c))\*\*0.5)/(2\*a)), *complex*((-b)/(2\*a), - (-(-(b\*\*2-4\*a\*c))\*\*0.5)/(2\*a)))

    print("The roots are complex.")

elif (b\*\*2-4\*a\*c) == 0:

    print((-b)/(2\*a))

    print("The roots are real and equal.")

else:

    print((-b + ((b\*\*2-4\*a\*c)\*\*0.5))/(2\*a), (-b - ((b\*\*2-4\*a\*c)\*\*0.5))/(2\*a))

    print("The roots are real and distinct.")