|  |  |  |  |
| --- | --- | --- | --- |
| Instructor |  | Due Date |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Part | **1** | **2** | **3** | **4** | Total |
| *Maximum Points* | **25** points | **25** points | **25** points | **25** points | **100**G101010 pointsG |
| ***Your Score*** |  |  |  |  |  |

**Textbook Reading Assignment**

Thoroughly read Chapter(s) 2 and 3 in your **Python** textbook.

**Part 1 Glossary Terms**

Define, in detail, each of these glossary terms from the realm of computer programming logic and design and computer topics, in general. If applicable, use examples to support your definitions. Consult your notes or course textbook(s) as references or the Internet by visiting Web sites such as:

[**http://www.ask.com**](http://www.ask.com) or [**http://www.webopedia.com**](http://www.webopedia.com/)

**(a) Argument ( function )**

|  |
| --- |
| An argument is what gets passed into a function, e.g., the string “bar” in `foo(“bar”)`. |

**(b) Comments**

|  |
| --- |
| Developers use comments (marked by a hash tag # in Python, // in JavaScript, etc.) to communicate to other developers what their code does, as well as to remind themselves what their code does if and when they need to dive back into it after some time has passed. |

**(c) Exponent Operator ( \*\* )**

|  |
| --- |
| Raises a number to the nth power; n\*\*power; 2\*\*3 = 8 |

**(d) format Function**

|  |
| --- |
| Formats a string in a print statement (made redundant by f strings, IMO) |

**(e) String Concatenation**

|  |
| --- |
| Joining strings together to form a longer string, either form variables or string literals. |

**Part 2 Textbook Exercises - Numerical Expressions**

**(1)** Determine the exact values of the following integer expressions.

**10.0** (a) 7 + 12 / 4 **7.25** (b) 7 + 3 / 12

**30** (c) 5 + 3 \* 10 + ( 8 - 13 ) **15.8333** (d) 10 + 4 / 8 + 16 / 3

**(2)** Determine the exact values of the following floating - point expressions.

**16.2**  (a) 10.0 + 4.0 / 20.0 + 18.0 / 3.0

**89.500** (b) ( 6.5 + ( 3.8 - 8.3 ) \* ( 5.0 - 13.5 )) \* 2.0

**(3)** Write a Python expression equivalent to 20 + 5 *A ­*− 4 *B* *C* .  
 **answer = 20 + (foo \* 5) – (4 \* bar \* fooBar)**

**(4)** Find the value of each of the following when r , s , t and u are integer variables initialized as r = 6 , s = 5 , t = 8 and u = 11 .

**30** (a) r = r \* 5  **15** (b) s = (s - 5) + 3 \* s

**-7.2** (c) t = t / 10 - 8  **12** (d) u = 2 \* u - 10

**(5)** Assume that one, two and three are decimal numbers with values 3.0 , 4.0 and  
 5.0 , respectively, and intFour, intFive and intSix are integers with values 9 , 10 and 11 , respectively. Find the value of each expression.

**(a)** two + three \* one

12.0

**(b)** intSix << 3

88

**(c)** intFour \* 2 / intFive \* 3

5.4

**(d)** intSix | intFour

11

**(e)** math.pow(two, 2)

16.0

**(f)** math.sqrt(two + one + 2)

3.0

**Part 3 Programming Exercises - Numerical Expressions**

For Questions **(1)** through **(7)**, choose from only the items given in the list below.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | |  | |  |  |  |
|  | **(a)** | primitive | **(e)** | def | **(i)** | double |  |
|  | **(b)** | apostrophe or single quote | **(f)** | memory | **(j)** | + and – |  |
|  | **(c)** | not | **(g)** | assert | **(k)** | float |  |
|  | **(d)** | 0 | **(h)** | lambda | **(l)** | print |  |
|  |  | |  | |  |  |  |

Using only the items above, fill in the following blanks with the most appropriate response by entering both the corresponding letter in the space provided below as well as the actual answer in the blank space, if present. There is only one valid response for each question.

**(1)** **e** The \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ keyword is used to create a new user defined function.

**(2) c** The \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ keyword is used to negate a Boolean value.

**(3) b** A(n) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ is used to delimit ( surround and distinguish ) character literals.

**(4) l** The \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ function displays output on the screen.

**(5) a** By default, real - valued literals are considered to be of type \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ .

**(6) f** The compiler needs to know the type of a data value so that the program can allocate \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ .

**(7) j** The \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ numeric operators can be applied to a single operand ( that is, they are unary operators ) .

**Part 4 Programming Exercises - Numerical Expressions**

**(1)** Write a complete Python programwhich prompts the user for a given number of seconds and then displays the number of hours, minutes and seconds equivalent to the given number of seconds.

For example, if the user enters 6320 , your program will display 1 hour, 45 minutes and 20 seconds, since 6320 seconds = 1 × 3600 + 45 × 60 + 20 × 1 .  
  
**def** extrapolateSeconds(seconds: int) -> str:

hours = int(seconds / 3600)

minutes = int(seconds / 60) % 60

secs = seconds % 60

**return** f"{seconds} seconds is {hours} hours, {minutes} minutes, {secs} seconds"

**print**(extrapolateSeconds(6320))