Name:_____

Please:

- Fill in the blanks, or circle best answer (for choices in round brackets)
- Give examples of the concept for blanks after "e.g."
- Type in the code in Spyder and test it out as needed!
- This exercise is not graded and not handed in.

1 Basic Concepts

- **Program**: Set of _____ written in a _____ (code), executed by the computer **exactly** as written.

- **Statement**: An _____, as part of a program, that performs a specific task (e.g._____, sets the value of x to 0.0, and _____ prints the value of x)

- **Algorithm**: Is a set of _____ _____ (e.g. _____)

- **Pseudo code**: Generic (non-language specific) code written in designing programs, useful for _____

- **Plot**: In the sense of scientific computing, a plot is _____ _____

2 Data

- **(Data) Type**: _____, e.g. float, int, string

- **Float**: A *data type* that represents _____ (e.g. _____)

- **Integer**: A *data type* that represents _____, (e.g. _____)

- **Boolean**: A *data type* that represents _____ (_____ or _____)

- **String**: A *data type* that represents _____ e.g. _____

- **Data structure**: _____ _____

- **List**: A **data structure**, surrounded by _____, are (mutable)/(immutable) and usually used for (heterogeneous)/(homogeneous) data, (e.g.: _____).

- **List element**: Each member of a **list**. For some list lst, its first element is accessed by lst_____, and its 3rd element by lst_____

- **Tuple**: *data structure* that is defined by a _____ but is often surrounded by _____, are (mutable)/(immutable) and can be used for (heterogeneous)/(homogeneous) data, e.g. _____

- **Array**: A *data structure* from the module _____, that stores (heterogeneous)/(homogeneous) data, can be multi-dimensional. Useful for working with scientific data (Faster)/(Slower) than lists.

3 Errors

- **Syntax error**: Error that arises from _____ in a program. E.g. `print('Hello'` is missing _____.

- **Type error**: Thrown when a variable used is the incorrect _____ e.g. _____

4 Program Flow

- **Control structure**: Statement or set of statements that _____ _____ e.g. _____

- **Conditional**: A control structure that _____, then performs different actions based on them e.g. _____

- **Relational operator**: _____ e.g. _____

- **Loop**: A *control structure* that _____ _____ E.g. _____

- **Iteration**: Is a(n) _____

- **Iterator variable**: Variable that _____. In the loop `for i in range(10):` the iterator variable is _____

5 Re-use of code

- **Function**: Arrangement of code that _____
  (e.g _____)

- **Defining a function**: Please write an example function definition that takes two input arguments, and returns the sum of the first squared and the second times 3.0

  _____

  _____

  _____

- **Calling a function**: Write an example of calling the function you've written, assigning its output to a variable:

  _____

  _____

  _____

  As a result of this, _____

- **Variable scope**: The area of code _____. Above, _____ are local to the function _____ (your function's name).

- **Module**: Group of _____ (e.g. _____, _____, _____)

- **Importing a module**: The best practice way is: _____, or _____.

6 Practice debugging

Please find the errors **without** attempting to run the program first!

6.1 Example 1

There are 3 errors in the following code, please correct them, and detail what the corrected code does

```
1 x = 15
2
3 if x < y:
4     print(x)
5 elif y < x
6     print(y)
7 else:
8     print('The two numbers are equal)
```

6.2 Example 2

Identify what type of error will occur with this code and state how this error could be corrected.

```
1  def hello():
2      """
3      A function that prints out 'Hello' x number of times
4      where x is given by the user. Returns none.
5      """
6      # Read in the integer x from the user
7      x = int(input('How many times you would like Hello printed?'))
8      print("x" * x)
9
10 hello()
```

6.3 Example 3

Identify the five errors in the following code, please identify and correct them

```
1  divisible():
2      """
3      A function that takes in two integers from the user,
4      x and y, and calculates whether x is divisible by y.
5      Returns none.
6      """
7
8      # Read in the integer x from the user
9      x = input('Please enter your first number: ')
10
11     # Read in the integer y from the user
12     y = input('Please enter your second number: ')
13
14     if x % y = 0:
15         print('{} is divisible {}'.format(x, y)
16     else:
17         print('{} is not divisible {}'.format(x, y))
18
19 divisible()
```

6.4 Example 4

Please discuss what the following function does:

```
1 def temp(L):
2     r = []
3     lsize = len(L)
4     while lsize > len(r):
5         element = L[-1]
6         L = L[:-1]
7         r.append(element)
8     return r
```

6.5 Example 5

There are 4 errors with the following code, please identify them

```
def birthdays(x):
    """
    A function that prints out the ages a group of x friends
    and the average age of the group. The ages are given
    by the user. Returns None.
    """

    # Initialise the list that you will populate
    Ages = []

    # Read in each friend's data
    count = 1  # count how many friends have entered their data

    while cout <= x:
        # Read in the ages from the user.
        d = int(input('Please enter your age '))

        # Append the new data to the list Ages
        Ages.append(d)


    print("\n The ages of your friends are:)

    # Iterate through list using indexing printing each element

    l = len(Age)
    for n in range(l):
        print(Ages[n])

    Average = sum(Ages)/len(Ages)

    print('The average age is {}'.format(Average))
```

Answers

1 Basic Concepts

- **Program**: Set of **statements** written in a programming language (code), executed by the computer exactly as written.
- **Statement**: An instruction, as part of a program, that performs a specific task (e.g.x = 0.0, sets the value of x to 0.0, ($x \leftarrow 0.0$) and print(x) prints the value of x)
- **Algorithm**: Is a set of step by step **instructions** that perform a designed task (e.g. root finding, numerical integration, etc.)
- **Pseudo code**: Generic (non-language specific) code written in designing programs, useful for useful for developing and communicating ideas
- **Plot**: In the sense of scientific computing, a plot is a graphical representation of numerical data

2 Data

- **(Data) Type**: Is a classification of data, e.g. float, int, string
- **Float**: A *data type* that represents real numbers (e.g. 3.14)
- **Integer**: A *data type* that represents integer numbers, (e.g. -1)
- **Boolean**: A *data type* that represents logical data (True or False)
- **String**: A *data type* that represents a group of characters e.g. "Hello World!"
- **Data structure**: Special arrangement of data that allows for ease of use, access and/or storage
- **List**: A **data structure**, surrounded by [ ] , are (mutable)/~~(immutable)~~ and usually used for ~~(heterogeneous)~~/(homogeneous) data, (e.g.: `[-1, 0, 1, 2]` or `['A', 'B', 'C', 'D']`).
- **List element**: Each member of a **list**. For some list L, its first element is accessed by L[0], and its 3rd element by L[2] (For L=[2.0, 4.0, 8.0, 16.0], L[0] == 2.0, and L[2] == 8.0)

- **Tuple**: *data structure* that is defined by a `,` but is often surrounded by `( )`, are ~~(mutable)~~/(immutable) and can be used for (heterogeneous)/~~(homogeneous)~~ data, e.g. (117, 'John')

- **Array**: A *data structure* from the module numpy, that stores ~~(heterogeneous)~~/(homogeneous) data, can be multi-dimensional. Useful for working with scientific data (Faster)/~~(Slower)~~ than lists.

3 Errors

- **Syntax error**: Error that arises from incorrect "language" or formatting in a program. E.g. `print('Hello'` is missing `)`.

- **Type error**: Thrown when a variable used is the incorrect type
  e.g. X=1.0/'CAT'

4 Program Flow

- **Control structure**: Statement or set of statements that tell a program when and how to perform certain actions e.g. **if**, statements **while** loops

- **Conditional**: A control structure that checks for boolean values of statements, then performs different actions based on them e.g. `if x > 0:  elif x == 0:  else:`

- **Relational operator**: Compares two objects (of the same type)
  e.g. `>, <, <=, >=, ==, !=`

- **Loop**: A *control structure* that repeats a block of code *for* a certain number of **iterations**, or *while* a condition is met.
  E.g. `while k > 0:`

- **Iteration**: Is a(n) individual repetition as part of a loop, or the process of looping

- **Iterator variable**: Variable that defines the current step of iteration within the loop.
  In the loop `for i in range(10):` the iterator variable is `i`

5 Re-use of code

- **Function**: Arrangement of code that accepts input, performs a task, and *often* returns output. (e.g `print()`, `numpy.sin()`, ...)

- **Defining a function**: Please write an example function definition that takes two input arguments, and returns the sum of the first squared and the second times 3.0

```
def new_fcn(x, y):
    z = x**2+3.0*y
    return z
```

- **Calling a function**: Write an example of calling the function you've written, assigning its output to a variable:

```
x1 = 3.0
y1 = 2.0
t = new_fcn(x1, y1)
```

As a result of this, $t == 15.0$

- **Variable scope**: The area of code for which a variable is defined. Above, `x, y, z` are local to the function `new_fcn` (your function's name).

- **Module**: Group of functions, classes, and variables in Python (e.g. numpy, scipy, matplotlib)

- **Importing a module**: The best practice way is: `import numpy as np`, or `import numpy`.

6 Practice debugging

Please find the errors **without** attempting to run the program first!

6.1 Example 1

There are 3 errors in the following code, please correct them, and detail what the corrected code does

```
1 x = 15
2 y = 10
3 if x < y:
4     print(x)
5 elif y < x:
6     print(y)
7 else:
8     print('The two numbers are equal')
```

Missing deceleration of y, (line 2) missing " : " (line 5), missing end quotation, line 8.

6.2 Example 2

Identify what type of error will occur with this code and state how this error could be corrected.

```
1  import sys
2  def hello():
3      """
4      A function that prints out 'Hello' x number of times
5      where x is given by the user. Returns none.
6      """
7      # Read in the integer x from the user
8      x = int(input('How many times you would like Hello printed?'))
9      print("Hello"*x)
10
11 hello()
```

The error is a *semantic* error, the letter "x" is printed x number of times, rather than "Hello".

6.3 Example 3

Identify the five errors in the following code, please identify and correct them

```
1  def divisible():
2      """
3      A function that takes in two integers from the user,
4      x and y, and calculates whether x is divisible by y.
5      Returns none.
6      """
7
8      # Read in the integer x from the user
9      x = int(input('Please enter your first number: '))
10
11     # Read in the integer y from the user
12     y = int(input('Please enter your second number: '))
13
14     if x % y == 0:
15         print('{} is divisible {}'.format(x, y))
16     else:
17         print('{} is not divisible {}'.format(x, y))
18
19 divisible()
```

- The function definition is incomplete (missing "**def**" on line 1)
- The input variables (x and y) are not converted to integers as indicated (lines 9 and 12)
- Assignment (=) has been used rather than check for equality (==) on line 14
- Missing close bracket on line 15.

6.4 Example 4

Please discuss what the following function does:

```
1  def temp(lst_in):
2      """
3      Takes a list as input, returns it in reverse order.
4      """
5      # Initialize a list
6      r = []
7
8      # Get original size
9      lsize = len(lst_in)
10
11     # Loop while output list is smaller than the input list
12     while lsize > len(r):
13         # lst_in[-1] refers to the last element in lst_in
14         element = lst_in[-1]
15
16         # Remove last element in the input list
17         lst_in = lst_in[:-1]
18
19         # Appends element from lst_in to the output list r
20         r.append(element)
21
22     # Finally return r
23     return r
```

As described in the docstring, this function reads in a list, and returns it in reverse order

6.5 Example 5

There are 4 errors with the following code, please identify them

```
1  def birthdays(x):
2      """
3      A function that prints out the ages a group of x friends
4      and the average age of the group. The ages are given
5      by the user. Returns None.
6      """
7
8      # Initialise the list that you will populate
9      Ages = []
10
11     # Read in each friend's data
12     count = 1  # count how many friends have entered their data
13
14     while count <= x:
15         # Read in the ages from the user.
16         d = int(input('Please enter your age '))
17
18         # Append the new data to the list Ages
19         Ages.append(d)
20         count += 1
21
22     print("\n The ages of your friends are:")
23
24     # Iterate through list using indexing printing each element
25
26     l = len(Ages)
27     for n in range(l):
28         print(Ages[n])
29
30     Average = sum(Ages)/len(Ages)
31
32     print('The average age is {}'.format(Average))
```

- Variable "cout" is not defined (should be *count*) on line 14
- `count` is not incremented within the loop
- Missing end quotation on line 22
- Variable "Age" is not defined on line 26, missing an "s".