In all questions, please choose the most appropriate answer if it appears that more than one match. All questions relate to the Python programming language. If an evaluation produces an error of any kind then choose Error as your answer. Different questions can have different numbers of choices.

1. Which of the following expressions evaluate to `10`?
   (a) 2 + 3 * 2
   (b) (2 + 3) * 2
   (c) 11 - 3 - 2
   (d) 25 - 4 / 2

2. What does the expression `(4-9)/3` evaluate to?
   (a) -2
   (b) -1
   (c) 2
   (d) Error

3. What does the expression `'Hello' - 'World'` evaluate to?
   (a) 'Hello World'
   (b) 'HelloWorld'
   (c) Error
   (d) None of the above.

4. What does the expression `2 + 'Brian'` evaluate to?
   (a) '2Brian'
   (b) '2 Brian'
   (c) 'BrianBrian'
   (d) Error

5. What does the expression `2*'Brian'` evaluate to?
   (a) '2Brian'
   (b) '2 Brian'
   (c) 'BrianBrian'
   (d) Error

6. After the assignment `x = 'Cheese Shop'`, which of the following assigns `'S'` to the variable `y`?

   (a) `y = x[7]`

   (b) `y = x[8]`

   (c) `y = x[-3]`

   (d) None of the above.

7. After the assignment `x = 'Cheese Shop'`, which of the following assigns `'ese'` to the variable `y`?

   (a) `y = x[3 : 5]`

   (b) `y = x[3 : 6]`

   (c) `y = x[4 : 5]`

   (d) `y = x[4 : 6]`

   (e) None of the above

8. After the assignment `x = 'Cheese Shop'`, which of the following assigns `'ese'` to the variable `y`?

   (a) `y = x[-8 : -5]`

   (b) `y = x[-8 : -6]`

   (c) `y = x[-7 : -5]`

   (d) `y = x[-7 : -6]`

   (e) None of the above

9. After the assignment `x = 'Cheese Shop'`, which of the following assigns `'S e'` to the variable `y`?

   (a) `y = x[7 : 3]`

   (b) `y = x[7 : 3 : -1]`

   (c) `y = x[7 : 4]`

   (d) `y = x[7 : 4: -1]`

   (e) None of the above

10. After the assignment `x = 'Cheese Shop sketch'`, what does the expression `x.split(' ')` evaluate to?

    (a) `['Cheese', 'Shop sketch']`

    (b) `['Cheese ', 'Shop sketch']`

    (c) `['Cheese', 'Shop', 'sketch']`

    (d) `['Cheese ', 'Shop ', 'sketch']`

    (e) Error

11. What is the value of `y` after the following is evaluated?

```
y = [11, 12, 13, 14, 15]
y.append(16)
y.append(17)
y.pop(len(y)-2)
```

    (a) `[11, 12, 13, 14, 15, 16]`

    (b) `[11, 12, 13, 14, 15, 17]`

    (c) `[11, 12, 13, 14, 16, 17]`

    (d) `[11, 12, 13, 15, 16, 17]`

    (e) Error

12. What is the value of `y` after the following is evaluated?

```
d = {}
d['Bill'] = 0
d['Bill'] = 1
d['Jill'] = 4
d['Fred'] = 5
d['Nobody'] = 99
y = d['Bill']
```

    (a) `0`

    (b) `1`

    (c) `[0,1]`

    (d) Error

13. What is the value of `y` after the following is evaluated?

```
d = {}
d['Bill'] = 0
d['Bill'] = 1
d['Jill'] = 4
d['Fred'] = 5
d['Nobody'] = 99
y = str(d['Julie'])
```

(a) `'99'`

(b) `'Julie'`

(c) `'None'`

(d) `'0'`

(e) Error

14. What is the value of `y` after the following is evaluated?

```
d = {}
d['Bill'] = 0
d['Bill'] = 1
d['Jill'] = 4
d['Fred'] = 5
d['Nobody'] = 99
y = str(d.get('Julie'))
```

(a) `'99'`

(b) `'Julie'`

(c) `'None'`

(d) `'0'`

(e) Error

The next 3 questions refer to the following definition.

```
def f(x,y):
    if x > y: z = x-y
    elif x < y: z = y-x-1
    return z
```

15. What is the value of `n` after the following is evaluated?

    ```
    n = f(5,5)
    ```

    (a) `5`
    (b) `-5`
    (c) `0`
    (d) Error

16. What is the value of `n` after the following is evaluated?

    ```
    n = f(f(2,3),f(4,2))
    ```

    (a) `1`
    (b) `-1`
    (c) `2`
    (d) `0`
    (e) Error

17. What is the value of `n` after the following is evaluated?

    ```
    x = 7
    y = 5
    n = f(y,x)
    ```

    (a) `-2`
    (b) `2`
    (c) `1`
    (d) `-1`
    (e) Error

The next 3 questions refer to the following definition.

```
def f(xs,n,m):
    r = 0
    while n < m:
        if xs[n] % 2 == 0: r += 1
        n += 1
    return r
```

18. What is the value of `x` after the following is evaluated?

    ```
    x = f([1,2,3,4], 0, 4)
    ```

    (a) `0`
    (b) `1`
    (c) `2`
    (d) `4`
    (e) Error

19. What is the value of `x` after the following is evaluated?

    ```
    x = f([1,2,3], 0, 5)
    ```

    (a) `0`
    (b) `1`
    (c) `2`
    (d) `4`
    (e) Error

20. What is the value of `x` after the following is evaluated?

    ```
    x = f([1,2,3,4], 5, 0)
    ```

    (a) `0`
    (b) `1`
    (c) `2`
    (d) `4`
    (e) Error

The next 3 questions refer to the following definition.

```
def z(n,ys):
    i = 0
    r = []
    for x in range(n):
        r.append((x, ys[i]))
        i += 1
    return r
```

21. What is the value of `zs` after the following is evaluated?

```
zs = z(3, [4,5,6])
```

    (a) `([0,1,2], [4,5,6])`
    (b) `[(0, 4), (1, 5), (2, 6)]`
    (c) `[(0, 6), (1, 5), (2, 4)]`
    (d) `[(4, 0), (5, 1), (6, 2)]`
    (e) Error

22. What is the value of `zs` after the following is evaluated?

```
zs = z(3, [4,5])
```

    (a) `([0,1,2], [4,5])`
    (b) `[(0, 4), (1, 5)]`
    (c) `[(0, 5), (1, 4)]`
    (d) `[(1, 4), (2, 5)]`
    (e) Error

23. What is the value of `zs` after the following is evaluated?

```
zs = z(2, [4,5,6])
```

    (a) `([0,1], [4,5,6])`
    (b) `[(0, 4), (1, 5)]`
    (c) `[(0, 6), (1, 5)]`
    (d) `[(0, 5), (1, 6)]`
    (e) Error

The next 2 questions refer to the following definition that contains missing statements. Recall that a semicolon is used to allow two commands to be put on one line.

```
def number_lines(filename):
    """ Return the list of pairs each consisting of the line number
    of the file (counting from 0) and the float in the line.

    number_lines(string) -> list<(int, float)>

    Precondition: filename is the name of a readable file
    that contains one float per line.

    """
    f = open(filename, 'r')
    # initialisation code

    for line in f:
        # body code

    close f
    return result
```

24. What is the required initialisation code?

    (a) `lineno = 0; result = 0`

    (b) `lineno = 1; result = 0`

    (c) `lineno = 0; result = []`

    (d) `lineno = 1; result = []`

25. What is the required body code?

    (a) `lineno += 1; result.append((lineno, line))`

    (b) `lineno += 1; result.append((lineno, float(line)))`

    (c) `result.append((lineno, line)); lineno += 1`

    (d) `result.append((lineno, float(line))); lineno += 1`

The next question refers to the following definition that is missing the test in an if statement.

```
def order_insert(x, xs):
    """ Insert x into the sorted list xs so that the order is maintained

    order_insert(X, list<X>) -> void

    Precondition: xs is a non-empty ordered list
    """

    start = 0
    end = len(xs)-1
    if x < xs[0]:
        end = 0
    elif x < xs[end]:
        while start+1 < end:
            # loop invariant: xs[start] <= x < xs[end]
            half = (start+end)/2
            if    :                    ## << missing test
                end = half
            else:
                start = half
    xs.insert(end, x)
```

26. What is a suitable test for the if statement?

    (a) `x > xs[half]`

    (b) `x >= xs[half]`

    (c) `x <= xs[half]`

    (d) `x < xs[half]`

    (e) None of the above

27. Which of the following best describes the definition of method m of a class C?

    (a) The definition of m has at least one argument, one of which is a reference to the object itself, typically called `self`. Reference is made to a data attribute in the body of the definition of m by using `self.` as a prefix.

    (b) The definition of m has at least one argument, one of which is a reference to the object itself, typically called `self`. Prefixing a name by `self.` is for referencing class variables, not data attributes.

    (c) The definition of m has at least one argument, the first of which is a reference to the object itself, typically called `self`. Reference is made to a data attribute in the body of the definition of m by using `self.` as a prefix.

    (d) The definition of m has at least one argument, the first of which is a reference to the object itself, typically called `self`. Prefixing a name by `self.` is for referencing class variables, not data attributes.

28. Which of the following is true of a method `m` of class `C`?

    (a) The method `m` must return a value.

    (b) The method `m` must either print something or return a value.

    (c) The method `m` must update at least one data attribute of `C`.

    (d) None of the above.

29. Given an object `x` of class X and an object `y` of class Y, what must be true in order that `x + y` be defined?

    (a) X and Y have to be the same class, `__add__` must be defined in that class (or super class) and it must return a value from the class.

    (b) X and Y have to be the same class, `__add__` must be defined in that class (or super class) and it must return a number.

    (c) X and Y can be different classes, `__add__` must be defined in class X (or a super class) and it must return a value from class X.

    (d) X and Y can be different classes, `__add__` must be defined in class X (or a super class) and it must return a value, but the value can be any type.

    (e) None of the above.

The 4 questions on the following page refer to the class definitions and assignments given below.

```
class A:
    def __init__(self, x):
        self.x = x

    def f(self, y):
        self.x += y

    def g(self):
        return self.x

    def __str__(self):
        return str(self.g())

class B1(A):
    def __init__(self):
        A.__init__(self, 0)

    def f(self, y):
        self.x = y**2

class B2(A):
    def __init__(self, x):
        A.__init__(self, x)

    def g(self):
        return -2*self.x

class C(B2):
    def __init__(self, x):
        B2.__init__(self, 2*x)

    def __str__(self):
        return str(-self.g())


a = A(2); a.f(3)
b1 = B1(); b1.f(3)
b2 = B2(2); b2.f(3)
c = C(2); c.f(3)
```

30. What does the expression `str(a)` evaluate to?

    (a) `'-10'`
    (b) `'5'`
    (c) `'9'`
    (d) `'14'`
    (e) Error

31. What does the expression `str(b1)` evaluate to?

    (a) `'-10'`
    (b) `'5'`
    (c) `'9'`
    (d) `'14'`
    (e) Error

32. What does the expression `str(b2)` evaluate to?

    (a) `'-10'`
    (b) `'5'`
    (c) `'9'`
    (d) `'14'`
    (e) Error

33. What does the expression `str(c)` evaluate to?

    (a) `'-10'`
    (b) `'5'`
    (c) `'9'`
    (d) `'14'`
    (e) Error

34. Which of the following best describes the purpose of the `Frame` widget in the tkInter GUI library?

    (a) It is responsible for the padding between widgets and for the order in which the widgets are placed.

    (b) It is responsible for the padding between widgets but not for the order in which the widgets are placed.

    (c) It is used as a container in which other widgets can be placed.

    (d) It is used to arrange a collection of widgets inside another widget.

35. What expression needs to replace `####` in order to complete the following recursive definition for summing the numbers from 1 to n.

```
def recursive_sum(n):
    """ Return 1+2+...+n
    recursive_sum(int) -> int
    Precondition n >= 1
    """
    if n == 1:
        return 1
    else:
        return ####
```

    (a) `1 + recursive_sum(n-1)`

    (b) `n + recursive_sum(n-1)`

    (c) `(n-1) + recursive_sum(n)`

    (d) `recursive_sum(n-1)`

    (e) None of the above

The next 2 questions refer to the following incomplete recursive function for computing all the permutations of a given list. An example of the result of calling the perms function is

```
perms([1,2,3]) == [[1,2,3],[2,1,3],[2,3,1],[1,3,2],[3,1,2],[3,2,1]]
```

Looking at the example above, the result is obtained by getting the permutations of `[2,3]` and for each of those adding 1 in all possible positions.

You may assume a function `shuffle` that takes a value and a list of lists and adds the value in all possible positions of each sublist of the list of lists. For example,

```
shuffle(1, [[2,3], [3,2]])
    == [[1,2,3],[2,1,3],[2,3,1],[1,3,2],[3,1,2],[3,2,1]]
```

The incomplete recursive definition is

```
def perms(xs):
    if xs == []: return [[]]
    # sub_perms is the list of all the permutations of xs with
    # the first element removed
    sub_perms = ## 1 ##
    return ## 2 ##
```

36. What expression should replace `## 1 ##`?

    (a) `perms(xs.pop(0))`

    (b) `perms(xs[1:])`

    (c) `perms(xs[1:-1])`

    (d) Any of the above.

37. What expression should replace `## 2 ##`?

    (a) `shuffle(xs[0], sub_perms)`

    (b) `perms(shuffle(xs[0], sub_perms))`

    (c) `shuffle(perms(xs[0]), sub_perms)`

    (d) None of the above.

38. What is the time complexity, in terms of the length of the list, of the following function that returns the tail of a list?

```
def tail(xs):
    """ Return the tail of the non-empty list xs
    tail(list<X>) -> list<X>
    """
    return xs[1:]
```

(a) Constant

(b) Logarithmic

(c) Linear

(d) Quadratic

(e) Exponential

39. What is the time complexity, in terms of the size of `n`, of the following function?

```
def f(n):
    t = 0
    m = 1
    while m <= n:
        p = 1
        while p <= m:
            t += p
            p += 1
        m += 1
    return t
```

(a) Constant

(b) Logarithmic

(c) Linear

(d) Quadratic

(e) Exponential

40. What is the value of `y` after the following has been evaluated?

```
g = lambda x:  'A' <= x <= 'Z'
f = lambda x: 'Spam'
y = [f(x) for x in 'Hello World' if g(x)]
```

(a) `['H', 'e', 'l', 'l', 'o', ' ', 'W', 'o', 'r', 'l', 'd']`

(b) `['H', 'W']`

(c) `['Spam']`

(d) `['Spam Spam']`

(e) `['Spam',  'Spam']`