THE UNIVERSITY OF QUEENSLAND
AUSTRALIA

This exam paper must not be removed from the venue

Venue ____________________

Seat Number ________

Student Number |__|__|__|__|__|__|__|__|

Family Name _____________________

First Name _____________________

School of Information Technology and Electrical Engineering

EXAMINATION

Semester One Final Examinations, 2015

CSSE1001 Introduction to Software Engineering I

This paper is for St Lucia Campus students.

Examination Duration: 120 minutes

Reading Time: 10 minutes

Exam Conditions:

This is a Central Examination

This is a Closed Book Examination - specified materials permitted

During reading time - write only on the rough paper provided

This examination paper will be released to the Library

Materials Permitted In The Exam Venue:

(No electronic aids are permitted e.g. laptops, phones)

Any unmarked paper dictionary is permitted

An unmarked Bilingual dictionary is permitted

Calculators - No calculators permitted

Materials To Be Supplied To Students:

1 x Multiple Choice Answer Sheet

Rough Paper

Instructions To Students:

Additional exam materials (eg. answer booklets, rough paper) will be provided upon request.

Answer all questions on the supplied True/False Multiple Choice Answer Sheet. Each question is worth one mark. Total marks: 40.

For Examiner Use Only

Question	Mark

Total ________

In all questions, please choose the most appropriate answer if it appears that more than one match. All questions relate to the Python 3 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. Each question is worth one mark.

1. What does the expression `2.0 + 11/2` evaluate to?

   (a) 6.5

   (b) 7.0

   (c) 7.5

   (d) 7

   (e) Error

2. What does the expression `2 * [’1’, ’3’]` evaluate to?

   (a) `[2, 6]`

   (b) `[’2’, ’6’]`

   (c) `[’1’, ’1’, ’3’, ’3’]`

   (d) `[’1’, ’3’, ’1’, ’3’]`

   (e) Error

3. What does the expression `’2’ + [’1’, ’3’]` evaluate to?

   (a) `[3, 5]`

   (b) `[’3’, ’5’]`

   (c) `[’21’, ’23’]`

   (d) `[’2’, ’1’, ’3’]`

   [✓ (e) Error]

4. What does the expression `list(’2’) + [’1’, ’3’]` evaluate to?

   can add list to list

   (a) `[3, 5]`

   (b) `[’3’, ’5’]`

   (c) `[’21’, ’23’]`

   [✓ (d) `[’2’, ’1’, ’3’]`]

   (e) Error

5. After the assignment `x = 'Very Naughty Boy'`, which of the following assigns `'N'` to the variable `y`?

   (a) `y = x[-10]`

   (b) `y = x[-12]`

   (c) `y = x[5]`

   (d) `y = x[6]`

   (e) More than one of the above is correct.

6. After the assignment `x = 'Very Naughty Boy'`, which of the following assigns `'Naug'` to the variable `y`?

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

   (b) `y = x[5:9]` start from first one to end but not include end

   (c) `y = x[6:9]`

   (d) `y = x[6:10]`

   (e) More than one of the above is correct.

7. After the assignment `x = 'Very Naughty Boy'`, which of the following assigns `'hty'` to the variable `y`?

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

   (b) `y = x[-8:-4]`

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

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

   (e) More than one of the above is correct.

8. After the assignment `x = 'Very Naughty Boy'`, which of the following assigns `'B y'` to the variable `y`?

   if the index is go backward, need to use -1 go backward one step

   (a) `y = x[-4:-5:-1]`

   (b) `y = x[-4:-6:-1]`

   (c) `y = x[-3:-5:-1]`

   (d) `y = x[-3:-6:-1]` d

   (e) More than one of the above is correct.

The next 2 questions refer to the following definition.

```
def g(x, z):
    x.append(z)
    return x
```

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

```
y = [1, 2, 3]
g(y, 4).extend(g(y[:], 4))
```

extend(g([1,2,3,4],4))

now y become [1,2,3,4]

   (a) `[1, 2, 3, 4]`

   (b) `[1, 2, 3, 4, 4]`

   (c) `[1, 2, 3, 4, 1, 2, 3, 4]`

   [✓ (d) `[1, 2, 3, 4, 1, 2, 3, 4, 4]`]

   (e) `[1, 2, 3, 4, 4, 1, 2, 3, 4, 4]`

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

```
y = [1, 2, 3]
g(y[:], 4).extend(g(y, 4))
```

this part is actually changing y

   [✓ (a) `[1, 2, 3, 4]`]

   (b) `[1, 2, 3, 4, 4]`

   (c) `[1, 2, 3, 4, 1, 2, 3, 4]`

   (d) `[1, 2, 3, 4, 1, 2, 3, 4, 4]`

   (e) `[1, 2, 3, 4, 4, 1, 2, 3, 4, 4]`

11. After the assignment `x = ’a,,b,c,d,\n’`, what does the expression `x.strip().split(’,’)` evaluate to?

    [✓ (a) `[’a’, ’’, ’b’, ’c’, ’d’, ’’]`]

    (b) `[’a’, ’’, ’b’, ’c’, ’d’, ’\n’]`

    (c) `[’a’, ’b’, ’c’, ’d’]`

    (d) `[’a’, ’b’, ’c’, ’d’, ’’]`

    (e) `[’a’, ’b’, ’c’, ’d’, ’\n’]`

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

```
d = {’Brisbane’: {2013:24.1, 2014:24.2},
     ’Adelaide’: {2012:22.1, 2013:22.6, 2014:22.8}}
y = d.get(’Brisbane’, {}).get(2012)
```

    (a) `None`

    (b) `{}`

    (c) `[24.1, 22.8]`

    (d) `{2013:24.1, 2014:24.2}`

    (e) Error

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

```
d = {’Brisbane’: {2013:24.1, 2014:24.2},
     ’Adelaide’: {2012:22.1, 2013:22.6, 2014:22.8}}
y = d.get(’Adelaide’, {}).get(2012)
```

    (a) `None`

    (b) `{}`

    (c) `22.1`

    (d) `{2012:22.1, 2013:22.6, 2014:22.8}`

    (e) Error

The next 3 questions refer to the following definition. Recall that `a // b` is integer division.

```
def md(x):
    a, b = x
    a, b = a // b, a % b
    return (a,b)
```

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

```
n = md(md((20,3)))
```

(a) `(0, 0)`

[✓ (b) `(3, 0)`]

(c) `(3, 1)`

(d) `(6, 2)`

(e) Error

md((20,3))
a = 20, b = 3
a = 20//3 = 6   b = 20%3 = 2
(a, b) = (6,2)

md(6, 2)
a = 6, b = 2
a = 6//2 = 3 b = 6%2 = 0

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

```
n = md(md((30,2)))
```

(a) `(0, 0)`

(b) `(3, 0)`

(c) `(3, 1)`

(d) `(6, 2)`

[✓ (e) Error]

md((30,2))
a = 30, b = 2
a = 30//2 = 15, b = 30%2 = 0

md(15,0)
a = 15, b = 0
error

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

```
n = md(md((3,20)))
```

[✓ (a) `(0, 0)`]

(b) `(3, 0)`

(c) `(3, 1)`

(d) `(6, 2)`

(e) Error

md((3,20))
a = 3, b = 20
a = 3//20 = 0, b = 3%20 = 3
md((0, 3))
a = 0, b = 3
a = 0//3 = 0, b = 0%3 = 0

The next 4 questions refer to the following definition.

```
def fi(xs, n, m):
    s = xs[n]
    r = [s]
    while n < m:
        if xs[n] > s:
            s = xs[n]
            r.append(s)
        n += 1
    return r
```

17. What is the value of x after the following is evaluated?

```
x = fi([2,6,7,5,3,1], 2, 7)
```

(a) []
(b) [7]
(c) [5, 3, 1]
(d) [7, 5, 3, 1]
[✓ (e) Error]

s = 7
r = [7]
while 2<7:
  if 7>7: False
n = 3
3<7:
  if 5>7: false
n = 4

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

```
x = fi([2,6,7,5,3,1], 2, -1)
```

(a) []
[✓ (b) [7]]
(c) [5, 3, 1]
(d) [7, 5, 3, 1]
(e) Error

s=7
r=[7]
while n<m
return r

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

```
x = fi([2,6,6,8,3,1], 0, 5)
```

(a) []
(b) [2]
[✓ (c) [2, 6, 8]]
[✓ (d) [4, 6, 6, 8]]
(e) Error

s=2
r=[2]
0<5:
if 2>2:
n+1=1
1<5:
if 6>2:
s = 6
r = [2,6]
n = 2
if 2<5
6>6
n=
  <5
  >6:
s=
r = [2,6,]
n=
  <5
  >
n=5
5<5

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

```
x = fi([1, 4, 2, 4, 5], 1, 5)
```

(a) [4, 2, 4, 5]
(b) [4, 4, 5]
[✓ (c) [4, 5]]
(d) [5]
(e) Error

s = 4
r=[4]
1<5
4>4
n=2
2<5
2>4
n=3
3<5
4>4
n=4
4<5

The next 3 questions refer to the following definition.

```
def hti(xs, v, h):
    i = h
    inc = 0
    n = len(xs)
    while xs[i] is not None:
        if xs[i] == v:
            return
        inc += 1
        i = (i+inc) % n
    xs[i] = v
```

i = 1, inc = 0, n = 5
if 5 == 8:
inc = 1, i = (1+1)%5 = 2
if 6 == 8:
inc = 2, i = (2+2)%5=4
if 7==8:
inc=3, i = (4+3)%5=2
if 6==8:
inc = 4, i=(2+4)%5=1
5==8:
inc=5, i = (1+5)%5=1
5==8
inc=6, i=(1+6)%5=2
6==8
inc=7, i=(2+7)%5=4

21. What is the result of the following evaluation? In particular, if the evaluation does not have an error, what will the value of `xs` be after the evaluation?

```
xs = [None, 5, 6, None, 7]
hti(xs, 8, 1)
```

(a) The evaluation terminates without error with `xs == [None, 5, 6, None, 7]`

(b) The evaluation terminates without error with `xs == [8, 5, 6, None, 7]`

(c) The evaluation terminates without error with `xs == [None, 5, 6, 8, 7]`

(d) The evaluation does not terminate

(e) An `IndexError` exception is raised.

22. What is the result of the following evaluation? In particular, if the evaluation does not have an error, what will the value of `xs` be after the evaluation?

```
xs = [2, 5, 9, 4, None, None, 7, None]
hti(xs, 11, 0)
```

(a) The evaluation terminates without error with `xs == [2, 5, 9, 4, 11, None, 7, None]`

(b) The evaluation terminates without error with `xs == [2, 5, 9, 4, None, 11, 7, None]`

(c) The evaluation terminates without error with `xs == [2, 5, 9, 4, None, None, 7, 11]`

(d) The evaluation does not terminate

(e) An `IndexError` exception is raised.

23. What is the result of the following evaluation? In particular, if the evaluation does not have an error, what will the value of `xs` be after the evaluation?

```
xs = [2, 5, 9, 4, None, None, 7, None]
hti(xs, 11, 3)
```

(a) The evaluation terminates without error with `xs == [2, 5, 9, 4, 11, None, 7, None]`

(b) The evaluation terminates without error with `xs == [2, 5, 9, 4, None, 11, 7, None]`

(c) The evaluation terminates without error with `xs == [2, 5, 9, 4, None, None, 7, 11]`

(d) The evaluation does not terminate

(e) An `IndexError` exception is raised.

The next 3 questions refer to the following definition that is missing three lines of code. The function `get_column_sums` below reads data from a CSV (Comma Separated Values) file and returns the list of sums for each column. We assume the file contains rows of floating point numbers separated by commas (and possibly including spaces) and each row has the same number of floats. Below is an example of such a file and the result of applying the function top that file.

The following is an example of a data file (`values.txt`).

```
1.2,  1 ,2.3, 1.4, 0.1
0.7,1.5, 1.2, 2.4, 0.1
2.1,0.7, 1.4, 2.0, 0.1
```

```
>>> get_column_sums('values.txt')
[4.0, 3.2, 4.9, 5.8, 0.3]
>>>
```

The definition of the `get_column_sums` function with three missing lines and the result of applying the completed function to the file is given below.

```
def get_column_sums(filename):
    fd = open(filename, 'r')
    data = []
    for line in fd:
        parts = line.split(',')
        line_data = []
        for p in parts:
            ## line 1 ##
        data.append(line_data)
    column_sums = []
    for index in range(len(data[0])):
        colsum = 0
        for row in range(len(data)):
            ## line 2 ##
        ## line 3 ##
    return column_sums
```

24. What is the required code for `## line 1 ##`?

    (a) `line_data.append(p)`
    (b) `line_data.append(float(p.strip()))`
    (c) `line_data.extend(p)`
    (d) `line_data.extend(float(p.strip()))`
    (e) More than one of the above is correct.

25. What is the required code for `## line 2 ##`?

    (a) `colsum = data[index][row]`
    (b) `colsum = data[row][index]`
    (c) `colsum += data[index][row]`
    (d) `colsum += data[row][index]`
    (e) More than one of the above is correct.

26. What is the required code for `## line 3 ##`?

    (a) `column_sums.append(colsum)`
    (b) `column_sums.extend(colsum)`
    (c) `column_sums + colsum`
    (d) `column_sums + [colsum]`
    (e) More than one of the above is correct.

The next three questions refer to the following partial definition of a `Player` class.

```
class Player(object):
    def __init__(self, name, health):
        self._name = name
        self._health = health

    def update_health(self, amount):
        """Update the players health by amount (may be negative)"""
        ## line 1 ##

    def get_health(self):
        """Return the players health."""
        ## line 2 ##
```

Assume that the following has been evaluated.

```
frodo= Player(’Frodo’, 10)
```

27. What is the required code for `## line 1 ##` so that the method satisfies the comment?

    (a) `health += amount`

    (b) `_health += amount`

    (c) `self.health += amount`

    (d) `self._health += amount`

    (e) More than one of the above is correct.

28. What is the required code for `## line 2 ##` so that the method satisfies the comment?

    (a) `print(self.health)`

    (b) `return self.health`

    (c) `print(self._health)`

    (d) `return self._health`

    (e) More than one of the above is correct.

29. Which of the following correctly updates the health by `-10` for the object `frodo`?

    (a) `frodo.update_health(-10)`

    (b) `update_health(frodo, -10)`

    (c) `frodo.update_health() += -10`

    (d) `update_health(frodo) +=  -10`

    (e) More than one of the above is correct.

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

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

    def f(self, x):
        return self.g(x)+1

    def g(self, x):
        return x+1

class B(A):

    def g(self, y):
        return y + self.x

class C(B):
    def __init__(self, x, y):
        super().__init__(x)
        self.y = y
        self.x += y

    def g(self, y):
        return y

    def f(self, x):
        return super().g(x) + self.y

class D(C):
    def __init__(self, x, y):
        super().__init__(x, y)
        self.y = y + x

    def g(self, x):
        return x + self.x

a = A(2)
b = B(2)
c = C(2, 2)
d = D(2, 0)
```

g(3) = 7

c.f(3)

c = C(2,2) - c.x = 4, c.y = 2

c.f(3)
g(3)+4 = (3+4) + 2 = 9

30. What does the expression `b.g(3)` evaluate to?

    (a) `5`
    (b) `6`
    (c) `7`
    (d) `9`
    (e) Error

31. What does the expression `a.f(3)` evaluate to?

    (a) `5`
    (b) `6`
    (c) `7`
    (d) `9`
    (e) Error

32. What does the expression `b.f(3)` evaluate to?

    (a) `5`
    (b) `6`
    (c) `7`
    (d) `9`
    (e) Error

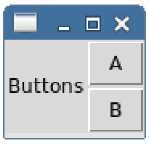
33. What does the expression `c.f(3)` evaluate to?

    (a) `5`
    (b) `6`
    (c) `7`
    [✓ (d) `9`]
    (e) Error

34. What does the expression `d.f(3)` evaluate to?

    (a) `5`
    (b) `6`
    (c) `7`
    (d) `9`
    (e) Error

The next two question relate to the following definitions. In our GUI application we decide we need a widget that contains two buttons and that this widget is to appear within the main window of the application to the right of the label as shown in the diagram below. This is what the application looks like at startup.



```
class ButtonsFrame(tk.Frame):
    def __init__(self,parent):
        super().__init__(parent.root)
        b1 = tk.Button(self, text= "A")
        b2 = tk.Button(self, text = "B")
        b1.pack()
        b2.pack()

class MainWindow(object):
    def __init__(self, root):
        self.root = root
        label = tk.Label(root, text="Buttons")
        label.pack(side=tk.LEFT,expand=1)
        bf = ButtonsFrame(self)
        bf.pack(side=tk.LEFT, fill=tk.BOTH,expand=1)
```

35. When the window is resized to make it larger, what will happen to the label?

    (a) The label will stay where it is relative to the top-left corner of the window.

    (b) The label will stay at the left but be centred vertically

    (c) The label will stay at the top but will share horizontal space with the buttons

    (d) The label will be centred vertically and share horizontal space with the buttons

36. When the window is resized to make it larger, what will happen to the buttons in the vertical direction?

    (a) Both buttons stay at the top (in their initial configuration)

    (b) Both buttons will stay together but be centered vertically

    (c) Button A will stay at the top with Button B centered vertically below Button A

    (d) Buttons A and B share the vertical space between them.

37. Below is a recursive definition of a (non-empty) linked list class. A linked list consists of a head (a value) and a tail (a linked list).

```
class LinkedList:
    def __init__(self, head, tail):
        self._head = head
        self._tail = tail

    def head(self): return self._head

    def tail(self): return self._tail

    def isLast(self): return self._tail == None

    def values(self):
        result, scan = [], self
        while not scan.isLast():
            result.append(scan.head())
            scan = scan.tail()
        result.append(scan.head())
        return result
```

Below is an incomplete **recursive** definition of `concat` for concatenating two linked lists to produce a linked list. As an example of its use consider the following involving the linked lists `x` and `y`.

```
>>> x.values()
[1, 2, 3, 4]
>>> y.values()
[5, 6, 7, 8]
>>> z = concat(x, y)
>>> z.values()
[1, 2, 3, 4, 5, 6, 7, 8]
```

What expression needs to replace `####` in order to complete the following **recursive** definition of `concat`?

```
def concat(xs, ys):
    if xs.isLast():
        return LinkedList(xs.head(), ys)
    else:
        return ####
```

(a) `LinkedList(ys.head(), concat(ys.tail(), xs))`

(b) `LinkedList(ys.tail(), concat(xs.head(), ys))`

(c) `LinkedList(xs.head(), concat(xs.tail(), ys))`

(d) `LinkedList(xs.tail(), concat(xs.head(), ys))`

(e) None of the above

38. What is the time complexity, in terms of the length of the list xs for the following function that reverses the elements of a list. You may assume inserting an element at position 0 is linear in the length of the list.

```
def reverse(xs):
    result = []
    for x in xs:
        result.insert(0, x)
    return result
```

(a) Constant
(b) Logarithmic
(c) Linear
(d) Quadratic
(e) Exponential

39. What is the time complexity, in terms of the length of the list xs for the following function that checks to see if xs has an increasing sublist of at least n elements. You may assume list indexing and calculating the length of a list are constant time and range is linear in the length of the range.

```
def contains_increasing(xs, n):
    """Returns True iff xs contains an increasing sublist
    with at least n elements

    Precondition: len(xs) > 2 and n > 1
    """
    v = xs[0]
    c = 1
    for i in range(1, len(xs)):
        if xs[i] > v:
            c += 1
            if c == n:
                return True
        else:
            c = 0
        v = xs[i]
    return False
```

(a) Constant
(b) Logarithmic
(c) Linear
(d) Quadratic
(e) Exponential

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

```
def t(xs):
    return [[xs[i][j] for i in range(len(xs))] for j in range(len(xs[0]))]

y = t([[1,2,3],[4,5,6]])
```

(a) `[1,2,3,4,5,6]`

(b) `[[1,2,3], [4,5,6]]`

(c) `[[1,4],[2,5], [3, 6]]`

(d) `[[1], [2], [3], [4], [5], [6]]`

(e) Error

END OF EXAMINATION