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 Two Final Examinations, 2016

CSSE7030 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

Instructions To Students:

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

Answer all questions on the supplied 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 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.

Question 1. What does the expression 3.0 + 5%3 evaluate to?

(a) 8.5

(b) 4.0

(c) 5.0

(d) 128

(e) None of the above

Question 2. What does the expression [′a′, ′b′] + list((1,2)) evaluate to?

(a) [’a’1, ’b’2]

(b) [’a’, ’b’, ’1’, ‘2’]

(c) [’a’, ’b’, [1, 2]]

(d) [’a’, ’b’, 1, 2]

(e) None of the above

Question 3. What does the expression 2 +2* [′2′, ′1′] evaluate to?

(a) [2,’4’, ’2’, ’4’, ’2’]

(b) [’2’,’4’, ’2’, ’4’, ’2’]

(c) [’2’,’2’, ’1’, ’2’, ’1’]

(d) [2,’2’, ’1’, ’2’, ’1’]

(e) Error

Question 4. What does the expression ′74′ + 2+′28′ evaluate to?

(a) [’74’, 2, ’28’]

(b) 84

(c) ’74228’

(d) [’74228’]

(e) Error

Question 5. What does the expression (9,2) + ((1)) evaluate to?

(a) (9, 2, 1)

(b) (9, 2, (1))

(c) (10, 3)

(d) (9, 3)

(e) Error

Question 6. After the assignment x = 'Sample text', which of the following assigns 'l' to the variable y?

(a) y = x[5]

(b) y = x[4:5]

(c) y = x[-6]

(d) y = x[6]

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

Question 7. After the assignment x = 'Sample text', which of the following assigns 'text' to the variable y?

(a) y = x[8 : 11]

(b) y = x[7 : 11]

(c) y = x[8 :]

(d) y = x[7 :]

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

Question 8. After the assignment x = 'Sample text', which of the following assigns 'ple' to the variable y?

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

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

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

(d) y = x[-7 : -5]

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

Question 9. After the assignment x = ‘Sample text’, which of the following assigns ’pet’ to the variable y?

(a) y = x[-8 : -4 : 2]

(b) y = x[-8 : -3 : 2]

(c) y = x[-7: -4 : 2]

(d) y = x[-7: -3 : 2]

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

The next 2 questions refer to the following definition.

```
def g(p):
    w = p.pop(1)
    p.extend(w)
    return p
```

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

```
y = [‘k’,’l’,’m’]
g(y[:]).extend(g(y))
```

(a) [‘l’,’m’,’k’]

(b) [‘k’,’m’,’l’]

(c) [‘l’,’m’,’k’,’l’,’m’,’k’]

(d) [‘k’,’m’,’l’,‘k’,’m’,’l’]

(e) Error

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

```
y = [‘k’,’l’,’m’]
g(y).extend(g(y[:]))
```

(a) [‘l’,’m’,’k’]

(b) [‘k’,’m’,’l’]

(c) [‘l’,’m’,’k’,’l’,’m’,’k’]

(d) [‘k’,’m’,’l’,‘k’,’l’,’m’]

(e) Error

Question 12. After the assignment z=’the big \n dog’ what does the expression sorted(z.split()) evaluate to?

(a) [‘big’, ‘dog’, ‘the’]

(b) [‘\n’, ‘big’, ‘dog’, ‘the’]

(c) ‘the big \n dog’

(d) [‘the’, ‘big’, ‘dog’]

(e) Error

Question 13. After the assignment z=’the big \n dog’ what does the expression sorted(z.split(sep=’ ’)) evaluate to?

(a) [‘big’, ‘dog’, ‘the’]

(b) [‘\n’, ‘big’, ‘dog’, ‘the’]

(c) ‘the big \n dog’

(d) ‘\n’, ‘big’, ‘dog’, ‘the’

(e) Error

Question 14. What is the value of a after the following is evaluated?

```
def f(x, y):
    y = y + [x]
    return y
a = [1,3]
a = f(2,a) + a
```

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

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

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

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

(e) None of the above

Question 15. What is the value of y after the following is evaluated?

```
d = {’aaa’: [1,2], ’bbb’:[3], ’ddd’:[]}
d[’ddd’] = [4]
y = d[’aaa’] + d[’bbb’] + d[’ccc’]
```

(a) [1, 2, 3]

(b) [1, 2, 4]

(c) [1, 2, None]

(d) Error

Question 16. What is the value of y after the following is evaluated?

```
d = {’aaa’: [1,2], ’bbb’:[3], ’ddd’:[]}
d[’ddd’] = [4]
y = d.get(’aaa’) + d.get(’bbb’) + d.get(’ccc’)
```

(a) [1, 2, 3]

(b) [1, 2, 4]

(c) [1, 2, None]

(d) Error

Question 17. What is the value of y after the following is evaluated?

```
d = {’aaa’: [1,2], ’bbb’:[3], ’ddd’:[]}
d[’ddd’] = [4]
y = d.get(’aaa’, []) + d.get(’bbb’, []) + d.get(’ccc’, [])
```

(a) [1, 2, 3]

(b) [1, 2, 4]

(c) [1, 2, None]

(d) Error

The next three questions refer to the following partial definition of a Student class for information about a student.

```
class Student(object):
    def __init__(self, name):
        self._name = name
        self._results = {}
def add_result(self, courseid, grade):
    """Add the grade for the given course to the results dictionary.
    add_result(str, int) -> None"""
    ## line 1 ##
def get_grade(self, courseid):
    """Return the grade for the given course. Raises an error if courseid is not present.
    get_grade(str) -> int"""
    ## line 2 ##
```

Assume that the following has been evaluated.

```
mbrown = Student("Mary Brown")
```

Question 18. What is the required code for ## line 1 ##?

(a) Student._results = {courseid:grade}

(b) Student._results[courseid] = grade

(c) self._results = {courseid:grade}

(d) self._results[courseid] = grade

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

Question 19. What is the required code for ## line 2 ##?

(a) print Student._results[courseid]

(b) return Student._results[courseid]

(c) print self._results[courseid]

(d) return self._results[courseid]

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

Question 20. Which of the following correctly updates the results for mbrown with a grade of 6 for ’CSSE1001’.

(a) add_result(mbrown, ’CSSE1001’, 6)

(b) add_result(mbrown)[’CSSE1001’] = 6

(c) mbrown.add_result(’CSSE1001’, 6)

(d) mbrown.add_result[’CSSE1001’] = 6

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

The next 3 questions refer to the following definition.

```
def m(x):
    a,b = x
    if a > b:
        return (b, a*a-b*b)
    elif a < b:
        return (a, b*b-a*a)
    else:
        return (a,b)
```

Question 21. What is the value of n after the following is evaluated? n = m(5,2)

(a) (6, -3)

(b) (4, 7)

(c) (6, 2)

(d) (3, 3)

(e) Error

Question 22. What is the value of n after the following is evaluated? n = m((4,3))

(a) (6, -4)

(b) (3, 7)

(c) (7, 2)

(d) (3, 6)

(e) Error

Question 23. What is the value of n after the following is evaluated? n = m(m((3,2)))

(a) (4, 16)

(b) (4, 18)

(c) (2, 21)

(d) (3, 19)

(e) None of the above

The next 3 questions refer to the following definition that is missing three lines of code. This function extracts URL information from the body of HTML text as in the example below.

```
>>> text = """
<body>
 <ul>
<li><a href="/future-students">Future Students</a></li>
 <li><a href="/current-students">Current Students</a></li>
</ul>
 </body>
"""
>>> get_urls(text)
 {‘Current Student’: ’/current-students’, ’Future Student’: ’/future-students’}
 >>>
```

The definition of the function get_urls is given below.

```
def get_urls(text):
    """Return a dictionary that associates each URL with the text for that URL
    get_urls(str) -> dict(str:str)
    Precondition: text is valid HTML """
    d = {}
    pos = 0
    next_pos = text.find("<a href")
    while next_pos != -1:
        ## line 1 ##
        end_tag = text.find(’>’, next_pos)
        start_tag = text.find(’<’, end_tag)
        ## line 2 ##
        d[name] = url
        ## line 3 ##
    return d
```

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

(a) url = text[next_pos:].split(’"’, 2)[0]

(b) url = text[next_pos:].split(’"’, 2)[1]

(c) url = text[:next_pos].split(’"’, 2)[0]

(d) url = text[:next_pos].split(’"’, 2)[1]

(e) None of the above

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

(a) name = text[end_tag:start_tag]

(b) name = text[end_tag+1:start_tag]

(c) name = text[end_tag:start_tag-1]

(d) name = text[end_tag+1:start_tag-1]

(e) None of the above

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

(a) break

(b) end_tag = text.find(’>’, start_tag)

(c) next_pos = text.find("<a href", start_tag)

(d) end_tag = text.find(’>’, next_pos)

(e) next_pos = text.find("<a href", next_pos)

Question 27. Which of the following statements about dictionaries is true?
(a) Dictionary keys can be any type.
(b) Only numbers or strings can be dictionary keys.
(c) Dictionary keys must be immutable types.
(d) Both dictionary keys and values must be immutable types.
(e) If the dictionary keys are mutable then the corresponding values must be immutable.

The next 3 questions refer to the following definition.

```
def f(xs):
    i = 0
    r = []
    while i >= 0 and i < len(xs):
        d,v = xs[i]
        r.append(v)
        i += d
    return r
```

Question 28. What is the value of zs after the following is evaluated?

```
zs = f([(1,'l'),(2,'m'),(3,'n'), (-1,'o')])
```

(a) [’l’, ’m’, ’n’, ’o’]

(b) [’l’, ’m’, ’o’, ’n’]

(c) [’l’, ’m’, ’n’]

(d) Non-terminating execution (possibly resulting in an out of memory error)

(e) Error (other than non-termination/out of memory)

Question 29. What is the value of zs after the following is evaluated?

```
zs = f([(1,’l’),(-2,’m’),(3,’n’), (-2,’o’)])
```

(a) [’l’, ’m’, ’n’, ’o’, ’n’]

(b) [’l’, ’m, ’n’]

(c) [’l’, ’m’]

(d) Non-terminating execution (possibly resulting in an out of memory error)

(e) Error (other than non-termination/out of memory)

Question 30. What is the value of zs after the following is evaluated?

```
zs = f([(1,’l’),(-1,’m’),(4,’n’), (-2,’o’)])
```

(a) [’l’, ’m’, ’n’, ’o’, ’n’]

(b) [’l’, ’m’, ’o’, ’n’]

(c) [’l’, ’m’, ’n’]

(d) Non-terminating execution (possibly resulting in an out of memory error)

(e) Error (other than non-termination/out of memory)

The next five questions refer to the class definitions and assignments given below.

```
class A(object):
    def __init__(self, x):
        self.x = x
    def g(self, x):
        return self.f(x)
    def f(self, x):
        return 2*x
class B(A):
    def g(self, y):
        return self.x + 3*y
class C1(B):
    def __init__(self, x, y):
        B.__init__(self, x)
        self.y = y
    def f(self, x):
        return self.x + self.y
class C2(B):
    def __init__(self, x, y):
        B.__init__(self, x)
        self.y = y
    def f(self, x):
        return x + self.x + self.y


a = A(3)
b = B(2)
c1 = C1(3, 5)
c2 = C2(3, 5)
```

Question 31. What does the expression b.g(2) evaluate to?

(a) 6

(b) 7

(c) 8

(d) 9

(e) None of the above

Question 32. What does the expression a.f(4) evaluate to?

(a) 5

(b) 6

(c) 7

(d) 8

(e) None of the above

Question 33. What does the expression b.f(5) evaluate to?

(a) 5

(b) 8

(c) 10

(d) 12

(e) None of the above

Question 34. What does the expression c1.g(2) evaluate to?

(a) 9

(b) 12

(c) 14

(d) 16

(e) None of the above

Question 35. What does the expression c2.f(4) evaluate to?

(a) 9

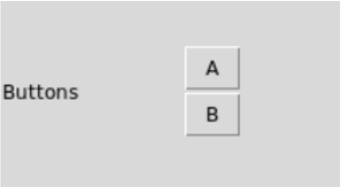
(b) 12

(c) 14

(d) 16

(e) None of the above

The next two questions relate to the following partial 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. (Recall that a ; can be used to put two program statements on the one line.)



```
class ButtonsFrame(tk.Frame):

    def __init__(self, parent):

        tk.Frame.__init__(self, parent._root)

        b1 = tk.Button(self, text= "A")

        b2 = tk.Button(self, text = "B")

        ## line 1 ##


class MainWindow(object):

    def __init__(self, root):

        self._root = root

        label = tk.Label(self._root, text="Buttons")

        label.pack(side=tk.LEFT)

        bf = ButtonsFrame(self)

        ## line 2 ##
```

Question 36. What is the required code for ## line 1 ##?

(a) b1.pack(side=tk.LEFT); b2.pack(side=tk.LEFT)

(b) b1.pack(expand=1); b2.pack(expand=1)

(c) b1.pack(fill=tk.BOTH); b2.pack(fill=tk.BOTH)

(d) b1.pack(); b2.pack()

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

Question 37. What is the required code for ## line 2 ##?

(a) bf.pack(side=tk.LEFT, expand=1)

(b) bf.pack(side=tk.LEFT, fill=tk.BOTH, expand=1)

(c) bf.pack(side=tk.LEFT)

(d) bf.pack(side=tk.LEFT, fill=tk.BOTH)

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

Question 38. What expression needs to replace #### in order to complete the following recursive definition for converting a list of binary bits into a positive integer.

```
def binary2int(digits):
    """ Return the integer represented by the binary digits.
    Example: binary2int([1,1,0,1]) = 13
    binary2int(list(int)) => int
    """
    if digits == []:
        return 0
    else:
        return ####
```

(a) digits[0] + 2*binary2int(digits[1:])

(b) digits[0] + 2*binary2int(digits[:-1])

(c) digits[-1] + 2*binary2int(digits[1:])

(d) digits[-1] + 2*binary2int(digits[:-1])

(e) None of the above

This question relates to the incomplete recursive function definition below that provides indexing into a nested list. The following gives examples of calling the function.

```
>>> nested = [[[1, 2], 3], 7, [4, [5, 6]], [8, 9, 10]]

>>> recursiveIndex(nested, [])

[[[1, 2], 3], 7, [4, [5, 6]], [8, 9, 10]]

>>> recursiveIndex(nested, [2,1,0])

5
```

The first example above shows what happens when the second argument (the indexing list) is empty. The second example indexes the 0’th element of the 1’th element of the 2’th element of nested.

Question 39. What expression needs to replace #### in order to complete the following recursive definition for indexing a nested list

```
def recursiveIndex(nested, indexes):

    """Return the element of the required sublist of the nested list

    (nested) at the position specified by indexes

    Precondition: indexes specifies a "valid" element of the nested list """

    if indexes == []:

        return nested

    else:

        return ####
```

(a) recursiveIndex(nested[indexes[0]], indexes[:-1])

(b) recursiveIndex(nested[indexes[0]], indexes[1:])

(c) recursiveIndex(nested[indexes[-1]], indexes[:-1])

(d) recursiveIndex(nested[indexes[-1]], indexes[1:])

(e) None of the above

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

```
g = lambda x,y: (x-y)

f = lambda x,y: y > x

xs = [0, 2, 4, 6]

 y = [g(x,y) for x in xs for y in xs if f(x,y)]
```

(a) [2, 4, 2, 6, 4, 2]

(b) [-2, -4, -6, -2, -4, -2]

(c) [-2, 2, -4, 4, -6, 6]

(d) [2, 2, -4, 4, -6, 6]

(e) Error

END OF EXAMINATION