

Desk Number _____

Student Number _____

Student Name _____

School of Science and Engineering

FINAL EXAMINATION

Semester 2, Aug, 2020

CSC1001 Introduction to Computer Science

Examination Duration: 120 minutes

Reading Time: 10 minutes

This examination has 3 questions.

Exam Conditions:

This is a FORMAL Examination

This is a RESTRICTED OPEN BOOK Exam. Maximum of one (1) sheet of handwritten or printed notes double sided are permitted

Materials Permitted In The Exam Venue:

Maximum of one (1) sheet of handwritten or printed notes double sided, a scientific calculator without the functionalities of programming and file storage and a paper-based dictionary are permitted. **NO OTHER MATERIALS PERMITTED**

Materials To Be Supplied To Students:

1 × 12 Page Answer Booklet, one answer sheet for multiple choice questions.

Materials To Be Handed in After Exam:

After exam, only answer sheet for multiple choice questions and the answer booklet are to be handed in.

Question 1. (15 × 2% = 30%)

Pick the correct option in each of the following sub-questions. Note that only ONE option is correct.

- 1.1) Hexadecimal number 3FA9.B and binary number 11010110.001 equal to decimal numbers:
- A. 16297.6875 and 428.125
 - B. 16553.6875 and 214.125
 - C. 16297.6875 and 214.125
 - D. 16553.625 and 428.125

Answer: C

- 1.2) Concerning Python language, which of the following statements is correct?
- A. The python interpreter is a tool to check grammar mistakes of python code
 - B. Object-Oriented Programming is a unique feature of Python
 - C. A program written in Python must be contained in a PY file, e.g. example.py
 - D. Python is a cross-platform language: Python's programs written on Windows can also run on Linux and macOS

Answer D

- 1.3) The following function returns the index of the minimum value in the a array. What should be filled in the comment line (#①) ?

```
a=[7, 8, 3, 1, 5]

def min_array(a):
    i=0
    min_idx=1
    for i in range(len(a)):
        if(a[i]<a[min_idx]):
            #① Please fill in code in this line
    return min_idx
```

- A: i=min_idx
- B: a[min_idx]=a[i]
- C: min_idx=a[min_idx]
- D: min_idx=i

Answer: D (Zhou Liguang)

1.4) Consider the following program, which statement/s is/are correct?

```
s1 = 'CSC'
s2 = ' '*3
s3 = '2020'
s4 = '3'

print(s1 + s2 + s3 + '\n' + s4)
print(s1 * s2 * s3 * s4)
print((s1 + s3 + s2[0:2]) * int(s4))
```

- A. The output of the first print function is
CSC 2020 3
- B. The output of the second print function does not exit.
- C. The output of the third print function is
CSC2020CSC2020CSC2020
- D. The output of the third print function does not exit.

Answer: B (Che Haoxuan, Modified by Li Yishu)

- 1.5) Concerning one python list, which of the following statement is correct?
- A. List is immutable, which means we cannot change its value using index operator.
 - B. List can contain another python list as its element.
 - C. Lists are comparable and two lists are equal if their first elements are equal.
 - D. `list(5)` creates a list containing a single element 5.

Answer: B (Li Zhaoqun)

1.6) Concerning data structure and algorithm, which of the following statements is incorrect?

- A. When evaluating an algorithm, we only need to measuring its running time.
- B. Data structure concerns how to organize and access data.
- C. An algorithm is a step-by-step procedure for performing some tasks in a finite amount of time.
- D. When analyzing the running time of an algorithm, two commonly used approaches are experimental analysis and asymptotic analysis.

Answer: A (Li Yishu)

1.7) Concerning the object in Python, which of the following statements is incorrect?

- A. Every object in Python has an unique ID.
- B. The type of an object cannot be determined automatically by Python interpreter, programmer must define its data type.
- C. Every variable is essentially a reference to an object.
- D. The ID of an object may not be changed during the execution of the program.

Answer: B (Li Yishu)

1.8) Concerning the following program, which of the following statements is correct?

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

class B:
    def __init__(self, b):
        self.b=b

m_list=[1, 2, 3]

obj_a=A(m_list)
obj_b=B(m_list)

m_list.reverse()
obj_a.a[0]=5

print(obj_b.b)
```

- A. The output of the statement `print(obj_b.b)` is [1, 2, 3].
- B. The output of the statement `print(obj_b.b)` is [5, 2, 3].
- C. The output of the statement `print(obj_b.b)` is [3, 2, 1].
- D. The output of the statement `print(obj_b.b)` is [5, 2, 1].

Answer: D (Zheng Yujian)

1.9) Concerning class inheritance in Python, which of the following statements is incorrect?

- A. Inheritance enables you to define a general class and later extend it to define more specialized classes.
- B. A subclass may inherit data fields and methods from its superclass.
- C. Inheritance models the is-a relationships, so all is-a relationships could be modelled using inheritance.
- D. A subclass is not a subset of its superclass, usually it contains more information and methods than its superclass.

Answer: C (Li Yishu)

1.10) What is the output of the following code?

```
class A:
    def __new__(self):
        print("A's __new__() is invoked")
    def __init__(self):
        print("A's __init__() is invoked")

class B(A):
    def __init__(self):
        print("B's __init__() is invoked")

class C(B):
    def __new__(self):
        print("C's __new__() is invoked")

def main():
    a=A()
    b=B()
    c=C()

main()
```

- A. A's __new__() is invoked
A's __init__() is invoked
A's __new__() is invoked
B's __init__() is invoked
C's __new__() is invoked
B's __init__() is invoked
- B. A's __new__() is invoked
A's __new__() is invoked
C's __new__() is invoked
- C. A's __new__() is invoked
A's __new__() is invoked
B's __init__() is invoked
C's __new__() is invoked
- D. A's __new__() is invoked
A's __init__() is invoked
B's __init__() is invoked
C's __new__() is invoked

Answer: B (Zhao Huan)

1.11) Concerning algorithm analysis, which of the following statements is not correct?

- A. The result of asymptotic analysis depends on the hardware and software environment where you perform the algorithm.
- B. An algorithm with a time complexity of $O(1)$ is asymptotically better than an algorithm with a time complexity of $O(n)$.
- C. We can use the big O notation to order classes of functions by asymptotic growth rate and it is useful in the analysis of algorithms.
- D. In algorithm analysis, we focus on the growth rate of the running time as a function of the input size.

Answer: A (Qiu Liangdong)

1.12) Concerning the following program

```
def square(n):  
    Slist1 = []  
    Slist2 = []  
    i = 0  
    for j in range(1,n):  
        Slist1.append(j**2)  
    while(True):  
        if Slist1[i]<n:  
            Slist2.append(Slist1[i])  
            i+=1  
            continue  
        else:  
            break  
    return Slist2  
    print(Slist2)  
square(100)
```

Which of the following statement/s is/are correct?

- A. Function `square()` has only one return value.
- B. Function `square()` uses a definite loop to return a list of numbers that are square less or equal to the input argument `n`.
- C. The length of the list `Slist2` has been set beforehand.
- D. The result of the program is shown as following:

[1, 4, 9, 16, 25, 36, 49, 64, 81]

[1, 4, 9, 16, 25, 36, 49, 64, 81]

Answer: B (Ms Qianyu Zhang, Modified by Li Yishu)

1.13) Concerning stack and queue, which of the following statement is correct?

- A. More than one element of a stack can be accessed simultaneously if necessary.
- B. Elements can be inserted at any time in a queue, but only the element that has been in the queue the shortest can be next removed.
- C. A stack can be accessed based on the “last in first out” principle and the time complexity of inserting an element into a stack is constant.
- D. The time complexity of removing an element from a stack or a queue is linear.

Answer: C (Zhao Weibing)

1.14) Concerning linked list, which of the following statement/s is incorrect?

- A. Singly linked list is a collection of nodes that collectively form a linear sequence.
- B. The tail node of a singly linked list has **None** as its next reference.
- C. A node in a double linked list usually contains two references pointing to its previous and next node.
- D. The time complexity of removing the tail node of a double linked list is quadratic.

Answer: D (Li Yishu)

1.15) Concerning the search and sort algorithms, which of the following statement is NOT correct?

- A. The time complexity of bubble sort is $O(n^2)$.
- B. Binary search algorithm can be applied on an unsorted sequence.
- C. The time complexity of quick sort is $O(n \log n)$.
- D. Binary search is usually more efficient than sequential search.

Answer: B (Zhao Huan)

Question 2. (10 × 4% = 40%)

Pick the correct option/s in each of the following sub-questions. Note that there may be MULTIPLE correct options for each sub-question(one, two, three or four correct options all possible). With any wrong options will get 0 point for that problem, while missing any correct options will get half(2 points).

2.1) Which of the following is/are the python reserved word?

- A. not
- B. fail
- C. True
- D. in

Answer: ACD (Zhao Huan)

2.2) Concerning recursion, which of the following statement/s is/are correct?

- A. When a function makes two or more recursive calls, this function does not belong to multiple recursion.
- B. A recursive algorithm should have one or more base cases which is or are non-recursive.
- C. All the recursive calls are executed successively, that is, one will not be executed until the previous one finishes completely.
- D. The base case in recursive algorithm can return `nothing`.

Answer: BD (Zhang Qianyu)

2.3) Which of the following statement/s is/are correct?

- A. If we open a file as `open(file, mode='r')`, we could write some texts in this file.
- B. If we open a file and don't need it any more, we'd better close it by `file.close()`.
- C. We could read file by many different mode like by line, by size or by whole lines.
- D. We could directly write text as many data fields like integer, boolean or float in a file.

Answer: BC (Che Haoxuan)

2.4) Given the program, which of the following statement/s is/are correct?

```
def function(input):  
    input = input * 2  
    return input  
  
output1 = function('32')  
output2 = function(32)  
output3 = function(32.0)
```

- A. The data types of output1, output2, output3 are: str, int, float
- B. output1 is exactly: 64
- C. output2 is exactly: 64
- D. output3 is exactly: 64

Answer: AC (Chen Yu'ang)

- 2.5) Concerning the following program, which of the following statement/s is/are correct?

```
class A:
    def __init__(self, i=0):
        self.i=i

    def m1(self):
        self.i+=1

class B(A):
    def __init__(self, j=0):
        super().__init__(3)
        self.j=j

    def m1(self):
        self.i+=1

def main():
    b=B()
    b.m1()
    print(b.i)
    print(b.j)

main()
```

- A. Class B() has two superclasses.
- B. In class B(), the initializer of class A() is accessed by calling function super().
- C. Class B() has two data fields.
- D. The output of b.i and b.j are 4 and 0.

Answer: BCD (Zhao Weibing)

- 2.6) Concerning the following program, which of the following statement/s is/are correct?

```
class A:
    def __init__(self, m):
        self.m = m
        self.__n = m + 1

    def display(self):
        print("%d %d" % (self.m, self.__n))

def output(m):
    m = m + 1
    return m

m = 1
a1 = A(m)
a1.display()

output(m)
a2 = A(m)
a2.display()
```

- A. The value of global variable m will initially be set as 1, and then changed to 2.
- B. Within the initializer of class A(), the function display() is invoked.
- C. Data field n of class A() can only be accessed within definition of class A().
- D. The output of this program is:

1 2

1 2

Answer: CD (Qiu Liangdong)

2.7) Concerning the following program, which of the following statement/s is/are correct?

```
class Staff:
    def __init__(self, ID=000000, name='name', gender='M'):
        self.ID=ID
        self.name=name
        self.gender=gender

    def getInfo(self):
        return 'Staff(%d,%s,%s)'%(self.ID, self.name, self.gender)

    def __getInfo(self):
        return 'Staff(%d,%s,%s)'%(self.ID, self.name, self.gender)

    def printStaff(self):
        print(self.getInfo())

class Teacher(Staff):
    def __init__(self, ID=000000, name='name', gender='M', major='Chinese'):
        super().__init__(ID, name, gender)
        self.major=major

    def __getInfo(self):
        return 'Staff(%d,%s,%s)'%(self.ID, self.name, self.gender)

    def getInfo(self):
        return 'Teacher(%d,%s,%s,%s)'%(self.ID, self.name, self.gender, self.major)

    def printTeacher(self):
        print(self.getInfo())

class SeniorTeacher(Teacher):
    def __init__(self, ID=000000, name='name', gender='M', major='Chinese', title='SeniorTeacher'):
        super().__init__(ID, name, gender, major)
        self.title=title

    def getInfo(self):
        return 'Teacher(%d,%s,%s,%s,%s)'%(self.ID, self.name, self.gender, self.major, self.title)

    def printSeniorTeacher(self):
        print(self.getInfo())

t=SeniorTeacher(123456, 'Lily')
t.printSeniorTeacher()
t.printTeacher()
```

A. The output for this program is:

```
Teacher(123456,Lily,M,Chinese,SeniorTeacher)
Teacher(123456,Lily,M,Chinese)
```

B. The `getInfo()` method in `SeniorTeacher()` class has been overridden.

C. The `__getInfo()` method in `Teacher()` class is inherited from `Staff` class.

D. The `t.printTeacher()` method is actually invoke `getInfo()` method in `SeniorTeacher()` class.

Answer: BD (Li Yishu)

2.8) Concerning the class in Python, which of the following statement/s is/are correct?

- A. In Python, everything (e.g. string, float number, list) is an object.
- B. A class can only inherit one class, which is called single inheritance.
- C. Dynamic binding decides which method is invoked at run-time when the method appears many times in the inheritance chain.
- D. Every instance of a subclass is also an instance of its superclass, and vice versa.

Answer: AC (Li Zhaoqun)

2.9) Concerning algorithm analysis, which of the following statement/s is/are incorrect?

- A. The big-Oh notation allows us to say that a function $f(n)$ is larger than or equal to another function $g(n)$ up to a constant factor when n is large enough.
- B. The big-Oh notation is usually used to characterize the running time of an algorithm in the asymptotic sense.
- C. Function $6n^3 + 2n^2 + 2n$ is $O(n^3)$.
- D. When we analyse an algorithm, we are usually interested in its average performance regardless of the input size.

Answer: AD (Zheng Yujian)

2.10) Let P stand for the push operation for stack and O stand for the pop operation for stack, what is the stack operation sequence that changes a string from " $3 * a + b / c$ " to " $3a * bc / +$ " ?

(For example, the operation sequence to change a string from " ABC " to " BCA " is PPOPOO)

- A. PPPOOOPPOPPPOO
- B. POPOPOPPPOPPPOO
- C. POPPOOPPOPOOPO
- D. POPPOOPPOPPPOO

Answer: D (Zhou Liguang)

Question 3. (6%+10%+14% = 30%)

Answer the following questions.

3.1) The following program is to find the greatest common divisor (gcd) of two integers.

Sample input numbers: 12 and 14

Sample output: 2

```
def gcd(a, b):  
    i=1  
    while(i<=a and i<=b):  
        #①  
    return gcd  
print(gcd(12, 14))
```

a) Please fill in the blank to implement the function required. (The fill-in codes could be more than one line)

b) Modify the program using recursive algorithm. The structure is provided as below. Please fill in the blanks.

```
def Recurgcd(a, b):  
    low = min(a, b)  
    high = max(a, b)  
  
    if low == 0:  
        #①  
    elif low == 1:  
        #②  
    else:  
        #③  
print(Recurgcd(12, 14))
```

c) Write the output for the given program if the input numbers are updated as (21, 28).

Answer:

a)

```
def gcd(a, b):  
    i=1  
    while(i<=a and i<=b):  
        if(a%i==0 and b%i == 0):  
            gcd=i;  
            i+=1  
    return gcd  
  
print(gcd(12, 14))
```

b)

```
def Recurgcd(a, b):  
    low = min(a, b)  
    high = max(a, b)  
  
    if low == 0:  
        return high ①  
    elif low == 1:  
        return 1 ②  
    else:  
        return Recurgcd(low, high%low) ③  
  
print(Recurgcd(12, 14))
```

b) 7

3.2) Concerning the following program

```
def quickSort(array, l, r):
    if l < r:
        q = partition(array, l, r)
        quick_sort(array, l, q-1)
        quick_sort(array, q+1, r)
    print('end')

def partition(array, l, r):
    x = array[r]
    i = l+1
    for j in range(l, r):
        #①
        array[i+1], array[r] = array[r], array[i+1]
    return #②

L = [6, 5, 3, 10, 2, 4]
quick_sort(L, 0, 5)
```

Answer the following questions:

- a) This program is to sort a given list with ascending order (the output of the above program is **[2,3,4,5,6,10]**). Use quick sort to implement the sort. Please fill the blank ①② to complete the program. (The fill-in codes could be more than one line)
- b) How many times the 'end' is printed in the program? Please draw the recursive trace for this program.

Answer:

a)

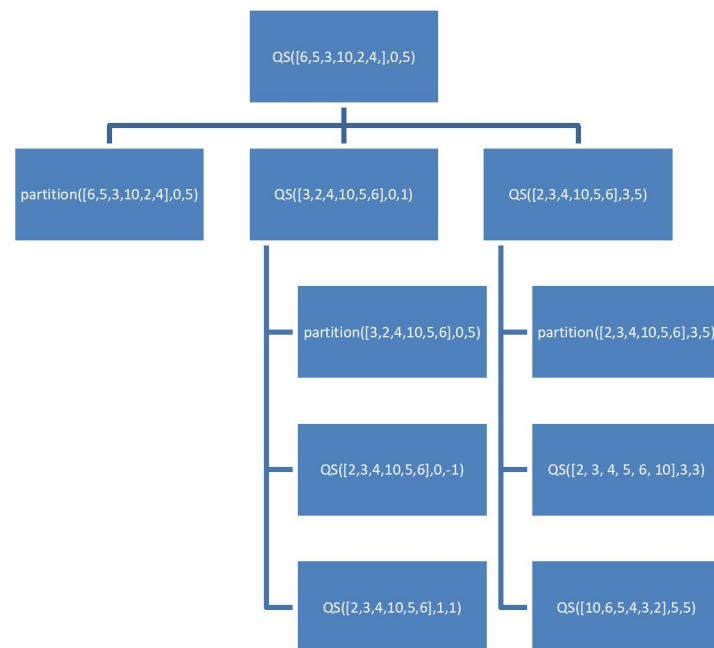
```
if array[j] <= x:
    i += 1
    array[i], array[j] = array[j], array[i]
```

①

② i+1

b)

3 times



- 3.3) Let a binary search tree (BST) is defined as follows:
 The left subtree of a node contains only nodes with elements less than the node's element.
 The right subtree of a node contains only nodes with elements greater than the node's element.
 Both the left and right subtrees must also be binary search trees.

a) What's the purpose of function *check()*.

```
class TreeNode:
    def __init__(self, e):
        self.element = e
        self.left = None
        self.right = None

def check(root):
    stack = []
    prev = None

    while root or stack:
        while root:
            stack.append(root)
            root = root.left
        root = stack.pop()
        if prev and root.element <= prev.element:
            return False
        prev = root
        root = root.right
    return True

root = TreeNode(4)
root.left = TreeNode(2)
root.left.left = TreeNode(1)
root.left.right = TreeNode(3)
root.right = TreeNode(6)
root.right.left = TreeNode(5)
root.right.right = TreeNode(7)
print(check(root))
```

b) Please write the output of above program.

c) The code below is to implement Depth First Search (DFS) algorithm on a binary tree. The input *t* is the reference of tree root node. Please write the output of applying DFS on the tree created in Question (a).

```
def DFSearch(t):
    if t:
        print(t.element)
        if (t.left is None) and (t.right is None):
            return
        else:
            if t.left is not None:
                DFSearch(t.left)
            if t.right is not None:
                DFSearch(t.right)

DfsSearch(root)
```


d) Write a function to find a node with maximum element from a binary search tree. (The fill-in codes could be more than one line)

```
def findMax(root):  
    if root.right is not None:  
        #①  
    else:  
        #②  
print(findMax(root).element)
```

Answer:

a) The function is to check whether a given a binary tree is a valid binary search tree (BST) or not.

b) True

c)

4
2
1
3
6
5
7

d)

```
def findMax(root):  
    if root.right is not None:  
        return findMax(root.right)  
    else:  
        return root  
print(findMax(root).element)
```

END OF EXAMINATION