

SCIT, University of Wollongong

CSIT110

Examination Paper

Session 3 2022

Student to complete:

Family name
Other names
Student number

--

Examination (60%) due on Wednesday 7th September 2022 at 2:15 PM

Exam duration: 3hours

Directions to students: Answer in the question booklet. 4 questions, each question is worth 15 marks

Question 1:

- A. Write two lines of code in the box below, one to instantiate a float variable, and another to instantiate a non-empty string variable. You may use any valid names for the variable names and assign the variables with any values of the specified data types. (1 mark)

Ans:

```
x = 1.0  
y = "hello world"  
# or any other acceptable statements
```

- B. What is the data type of the variable, x in the code below? (1 mark)

```
1. x = 3 + 1.5/5*2-1
```

Ans:

x is a float type object.

- C. What data type does the function input() return? (1 mark)

Ans:

string type

- D. Write a line of code to send a newline to the console. (1 mark)

Ans:

```
print()
```

E. What are the data types for the following variables? (2 marks)

1.	<code>w = 32</code>
2.	<code>x = w > 500</code>
3.	<code>y = ["a", "b", True, 2+3j]</code>
4.	<code>z = y[0]</code>

Ans: (you can spell the class names either in full or in short)

w: integer / int
x: boolean / bool
y: list
z: string / str

F. What is the output of the following code? (2 marks)

1.	<code>for num in range(0, 11, 2):</code>
2.	<code> print(num*1.1)</code>

Ans:

0.0
2.2
4.4
6.6
8.8
11.0

Given the following variable, answer the next two questions,

```
1. web = {  
2.     "title": "Welcome to my homepage",  
3.     "elements": {  
4.         "forms" : [],  
5.         "page_no" : 3,  
6.         "body" : {}  
7.     }  
8. }
```

G. How many key-value pairs are there in the variable `web`? (1 mark)

Ans:

2

H. Write one or two lines of code to excess the value mapped to the key `"page_no"` in the variable `web` and assign it to a variable named `page_num` (2 marks)

Ans:

```
page_num = web["elements"]["page_no"]
```

I. Given the following variables, what the length of the string sent to the console (2 marks)

```
1. x = 3  
2. y = 5  
3. z = 10  
4. print(f"a{x:<8}{y:^3}b{z:>2}c")
```

Ans:

16

J. What will be displayed on the console of the following code? Tick the correct checkbox. (2 marks)

```
1. print(type([]) is list)
```

☒ True ☐ False

Question 2:

Given the following function definition, answer the next two questions,

```
1. def filter(arg: list, threshold= 0.5):
2.     new_list = []
3.     for ele in arg:
4.         if ele >= threshold:
5.             new_list.append(ele)
6.         else:
7.             new_list.append(0)
8.     return new_list
9.
10. x1 = filter([0.1, 0.46, 0.78, 9.3])
```

A. What would the value of x1 be? (3 marks)

Ans:

[0, 0, 0.78, 9.3]

B. If the following line of code is run, a TypeError will be raised in line 4. Explain what causes the error? (3 marks)

```
11. x2 = filter([x1, 1, 2.3, 9.2])
```

Ans:

x1 is a list while threshold is a float. A comparison cannot be performed between different data type

C. What will be displayed on the console of the following code? Tick the correct checkbox. (3 marks)

```
1. s = 'foo'
2. t = 'bar'
3. print('barf' in 2 * (s + t))
```

☒ True ☒ False

D. Which of the following are valid ways to specify the string literal `csit'110` in Python? Check all that applies. (3 marks)

- ☒ `"csit'110"`
- ☐ `'csit''110'`
- ☐ `'csit'110'`
- ☒ `'csit\'110'`
- ☒ `"""csit'110"""`

E. Syntax error in python is detected by _____ at _____. Check the correct answer. (3 marks)

- ☐ – compiler ... run time
- ☐ – interpreter ... compile time
- ☒ – interpreter ... run time
- ☐ – compiler ... compile time

Question 3:

- A. What will be displayed on the console of the following code? Tick the correct checkbox. (3 marks)

```
1. myvar = [1, 'John', 1+3j]
2. print(type(myvar[2:3]))
```

- ☒ `<class 'list'>`
☐ `<class 'complex'>`
☐ `NameError`

Given the following function definition,

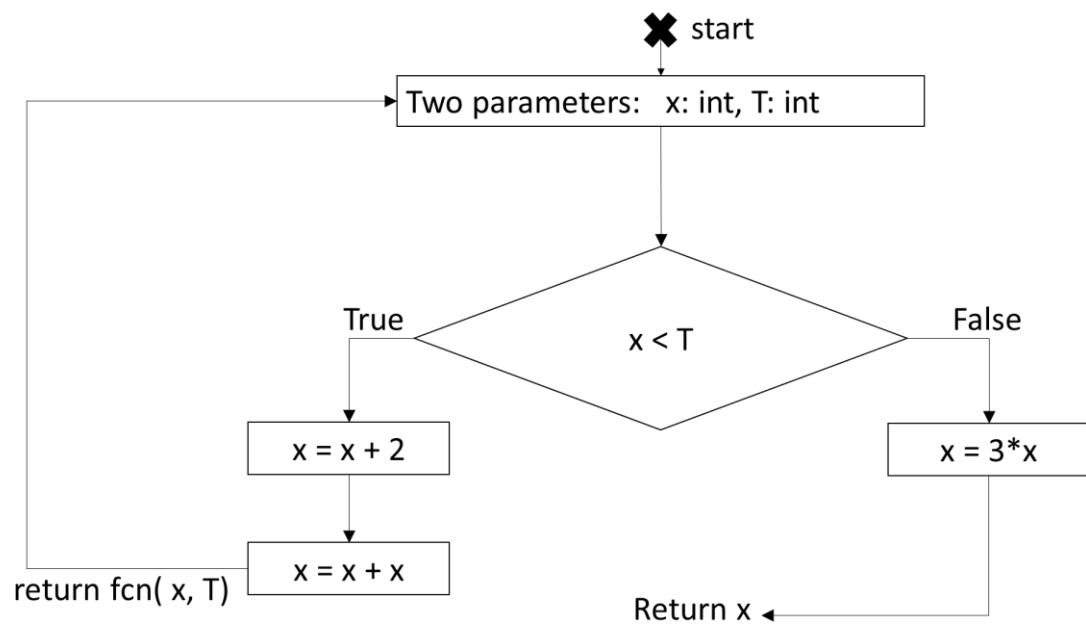
```
1. class SavingsAccount:
2.     rate = 0.002
3.     def __init__(self, amount = 5):
4.         self.balance = amount
5.
6.     def get_balance(self):
7.         return self.balance
8.
9.     def deposit(self, amount):
10.        self.balance += amount
```

- B. Write three lines of code to create an object from the class with an initial balance of \$100.00 and make a deposit of \$20.00. Send the balance of the object to the console with a \$ in the front and in 2 decimal places. (3 marks)

Ans:

```
obj = SavingsAccount(100)
obj.deposit(20)
print(f"${obj.balance:.2f}")
```

C. Define a function named, fcn, with the control flow depicted in the following diagram.



Ans: Use an arrow \rightarrow to depict each level of indentation in a line (4 marks)

```
def fcn(x, T):
    →if x < T:
        →→x += 2
        →→x += x
        →→return fcn(x, T)
    →else:
        →→return 3*x
```

D. Given the function definition on the next page, what is displayed on the console?
(4 marks)

Ans:

```
4
3
```



```

class PriceTag:
    tax = 0.07
    def __init__(self, value: float):
        self.value = value
    def total(self, with_tax = True):
        if with_tax:
            return self.value*(PriceTag.tax+1)
        else:
            return self.value

    def add(self, price: 'PriceTag'):
        """ the argument price is a PriceTag object """
        self.value += price.value
    def __str__(self):
        return f"${self.total():.2f}"

class Product():
    stock = 0
    def __init__(self, price: float, brand: str):
        self.price = PriceTag(price)
        self.brand = brand
        Product.stock += 1

class Cart():
    num_item = 0
    def __init__(self):
        self.items = []
        self.total = PriceTag(0)
    def checkout(self):
        return str(self.total)

    def add_item(self, item: Product):
        self.items.append(item)
        self.total.add(item.price)

milk = Product(1.5, "HL")
eggs = Product(3.5, "Swee Choon")
rice = Product(11.9, "Royal Umbrella")
cheese = Product(5.60, "Nestle")
mycart = Cart()
mycart.add_item(milk)
mycart.add_item(eggs)
mycart.add_item(rice)
print(Product.stock)
print(len(mycart.items))

```

Question 4

This is a student's solution for the questions in the next page.

```
1. def Property: # q4a -> class Property: (2 marks)
2.     TYPE = "public"
3.     def __int__(-> __init__( (2 marks)
4.         self, num_of_units, psqm, area_sqm, tenure=99):
5.         self.num_of_units = num_of_units
6.         self.psqm = psqm
7.         self.area_sqm = area_sqm
8.         self.tenure = tenure
9.
10.     #classmethod @classmethod (2 marks)
11.     def from_dict(cls, pram) # q4b missing colon (2 marks)
12.         obj = cls(
13.             pram["num_of_unites"],
14.             pram["psqm"],
15.             pram["area_sqm"],
16.             pram["tenure"])
17.         return obj
18.
19.     def psf(self): # q4c
20.         self.psqm/10.7639 missing return key word (2 marks)
21.
22.     def get_price(self): # 4d
23.         price = Property.area_sqm to sqft(
24.                 self.area_sqm)*self.psf()(5 marks)
25.         return price
26.
27.     @staticmethod
28.     def area_sqm_to_sqft(area):
29.         return area*10.7639
```

- i. Spot five errors in the code and write the correct line beside or above it.
- ii. Fill in the missing lines of code

Reminder: the questions to the above solution is on the next few pages.

Q4a

Define a class that fulfils the following specifications.

Class name	Property
Parameter(s)/ argument(s) for the constructor, not inclusive of those that are automatically passed in.	<p>The constructor of this class takes in 4 parameters in this order,</p> <ol style="list-style-type: none"> 1. An int, <code>num_of_units</code> 2. A float, <code>psqm</code> 3. An int, <code>area_sqm</code> 4. An int, <code>tenure</code> <p>The last parameter, <code>tenure</code>, has a default value of 99</p>
Class Attribute(s)	<p>This class has 1 class attribute, <code>TYPE</code>, in uppercase. Assign the string object <code>'public'</code> to the class attribute.</p>
Instance Attribute(s)	<p>Every instance of this class has 4 instance attributes,</p> <ul style="list-style-type: none"> <code>num_of_units</code>, an integer object <code>psqm</code>, a float object <code>area_sqm</code>, a float object <code>tenure</code>, an integer object <p>At creation time (in the constructor), assign the parameters of the constructor to the instance attributes of the same name.</p>
Example usage	<pre>blk1 = Property(105, 250.5, 2005.5) print(blk1.num_of_units) print(blk1.psqm) print(blk1.area_sqm) print(blk1.tenure) print(Property.TYPE)</pre>
console display for example code	<pre>105 250.5 2005.5 99 public</pre>

Q4b Define a **class** method for the `Property` class.

Method name	<code>from_dict</code>
Method type	This is a class method
Parameter(s)/ argument(s)	<p>When the method is called, 1 argument must be provided.</p> <ol style="list-style-type: none"> 1. A dictionary object <p>Assume that the dictionary objects contain the following keys <code>"num_of_units"</code>, <code>"psqm"</code>, <code>"area_sqm"</code>, <code>"tenure"</code>.</p>
Return value	<p>This method returns 1 object.</p> <ol style="list-style-type: none"> 1. A Property object
Detailed description	<p>Using the values in the dictionary parameter, create and return an instance of the Property class.</p> <p>Your solution should map the correct values based on its keys to the parameters of the class constructor with the same name.</p>
Example usage	<pre>blk104 = Property.from_dict({ "num_of_units": 200, "psqm": 300.5, "area_sqm": 1002, "tenure": 103 }) print(blk104.num_of_units) print(blk104.psqm) print(blk104.area_sqm) print(blk104.tenure) print(Property.TYPE)</pre>
console display for example code	<pre>200 300.5 1002 103 public</pre>

Q4c Define an **instance** method for the **Property** class.

Method name	psf
Method type	This is an instance method
Parameter(s)/ argument(s), not inclusive of those that are automatically passed in.	When the method is called, no argument is required.
Return value	This method returns 1 object. 1. A float object
Detailed description	Return the instance attribute, <code>psqm</code> , divided by 10.7639. You do not need to format the float.
Example usage	<pre>blk1 = Property(105, 250.5, 2000) print(blk1.psf())</pre>
console display for example code	23.27223404156486

Q4d Define an **instance** method for the **Property** class.

Method name	get_price
Method type	This is an instance method
Parameter(s)/ argument(s), not inclusive of those that are automatically passed in.	When the method is called, no argument is required.
Return value	This method returns 1 object. 1. A float object
Detailed description	Return the price of the Property object. The price of the property is the product of the area in sqft and the psf. You do not need to format the float . Remember that psf is an instance method, not an instance attribute, remember to call the method!
Example usage	<pre>blk1 = Property(105, 250.5, 2000) print(blk1.get_price())</pre>
console display for example code	501000.0

Q4e Define a **static** method for the **Property** class.

Method name	area_sqm_to_sqft
Method type	This is a static method
Parameter(s)/ argument(s), not inclusive of those that are automatically passed in.	When the method takes in 1 parameter. 1. A float object, <code>area</code>
Return value	This method returns 1 object. 1. A float object
Detailed description	Return parameter, <code>area</code> , multiplied by 10.7639. You do not need to format the float .
Example usage	<code>print(Property.area_sqm_to_sqft(57))</code>
console display for example code	613.5423

-----End of Paper-----