School of Computing and Information Technology

Student to complete:

Family name	
Other names	
Student number	
Table number	

CSIT110
Fundamental Programming in Python
Singapore Institute of Management

Supplementary Examination Paper
Session 3 2022

Exam duration	3 hours
Weighting	60% of the subject assessment
Marks available	60 marks
Items permitted by examiner	Non-programmable calculator
Directions to students	4 questions to be answered. Marks for each question are shown beside the question. All answers must be written in the QUESTION BOOKLET provided.

This exam paper must not be removed from the exam venue

4 questions worth 15 marks each (Total 60 Marks)

For each of the following questions you should provide a solution in the blanks or in the boxes provided. The size of the blanks and boxes do not generally indicate the size of an appropriate answer.

Question 1:

A. Write two lines of code in the box below, one to instantiate a positive (>0) integer variable, and another to instantiate a string variable with 5 chars. You may use any valid names for the variable names and assign the variables with any values that meet the criteria of the specified data types. (1 mark)

Ans:

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

```
1.  x = input("enter an integer") + "100400"
```

Ans:

x is a _____ type object.

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

Ans:

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

Ans:

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

```
1. w = 32.5*1
2. x = w > 500
3. y = ["a", "b", True, 2+3j]
4. z = y[0]
```

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

w:

x:

y:

z:

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

```
1. for num in range(0, 5, 1):
2.     print(num*2.2)
```

Ans:

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:

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:

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:

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:

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:

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

```
1.  s = 'chocolate'
2.  t = 'bar'
3.  print('barc' 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'>
- ☐ IndexError

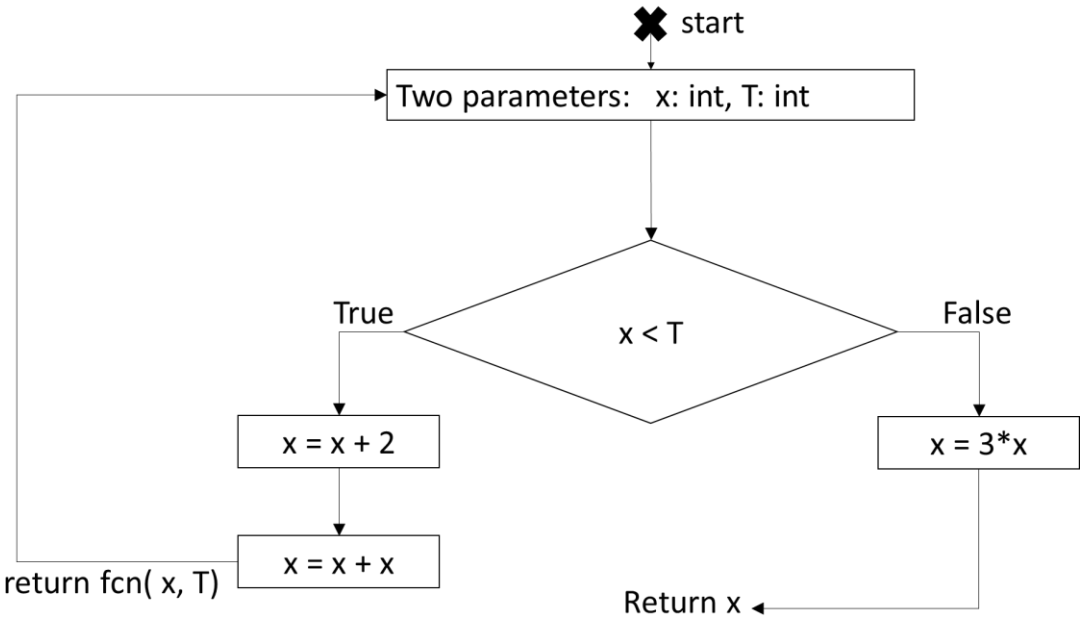
Given the following function definition,

```
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. (4 marks)

Ans:

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



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

Given the following program,

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

class Cart():
    num_item = 0
    def __init__(self):
        self.items = []
        self.total = 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")
apples = Product(5.0, "Royale")
mycart = Cart()
mycart.add_item(milk)
mycart.add_item(eggs)
mycart.add_item(rice)
print(Product.stock)
print(len(mycart.items))
```

D. What is displayed on the console?
(4 marks)

Ans:

Question 4

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

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

a. Spot five errors in the code and write the correct line beside or above it. (10 marks)
b. Fill in the missing line of code (5 marks)

Reminder: the specification to the above code is shown on the next few pages. You only need to answer questions a and b above.

i. 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.	The constructor of this class takes in 4 parameters in this order, 1. An **`int`**, `num_of_units` 2. A **`float`**, `psqm` 3. An **`int`**, `area_sqm` 4. An **`int`**, `tenure` The last parameter, `tenure`, has a default value of 99
Class Attribute(s)	This class has 1 class attribute, `TYPE`, in uppercase. Assign the **`string`** object `'public'` to the class attribute.
Instance Attribute(s)	Every instance of this class has 4 instance attributes, `num_of_units`, an **`integer`** object `psqm`, a **`float`** object `area_sqm`, a **`float`** object `tenure`, an **`integer`** object At creation time (in the constructor), assign the parameters of the constructor to the instance attributes of the same name.

Example usage	`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)`

console display for example code	105 250.5 2005.5 99 public

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

Method name	`from_dict`
Method type	This is a **`class`** method
Parameter(s)/ argument(s)	When the method is called, 1 argument must be provided. 1. A **`dictionary`** `object` Assume that the dictionary objects contain the following keys `"num_of_units"`, `"psqm"`, `"area_sqm"`, `"tenure"`.
Return value	This method returns 1 object. 1. A **`Property`** object
Detailed description	Using the values in the dictionary parameter, create and return an instance of the **`Property`** class. Your solution should map the correct values based on its keys to the parameters of the class constructor with the same name.

Example usage	`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)`

console display for example code	200 300.5 1002 103 public

iii. 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, `psqm`, divided by 10.7639. You do not need to format the float.
Example usage	`blk1 = Property(105, 250.5, 2000)` `print(blk1.psf())`
console display for example code	23.27223404156486

iv. 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	`blk1 = Property(105, 250.5, 2000)` `print(blk1.get_price())`
console display for example code	501000.0

v. 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, `area`
Return value	This method returns 1 object. 1. A **`float`** object
Detailed description	Return parameter, `area`, multiplied by 10.7639. You do not need to format the **`float`**.
Example usage	`print(Property.area_sqm_to_sqft(57))`
console display for example code	`613.5423`

End of Examination