

# Exam

## Object Oriented Programming 1 - Python

### BTH000

Time allowed: Three hours.

No books or notes are allowed by the students.

(International College, ZJUT, provides an English-Chinese dictionary in the exam room)

***Total Points: 100***

### ***Important!***

- All questions are related to Python Programming
- The questions are not ordered by difficulty, so if you get stuck on one question please go ahead with the next. You can always go back if there is time.

***Good Luck!***

## 1. SHORT ANSWERS

 $\Sigma$ : 21

- (a) Name four different built-in types in Python, that we covered in this course. (2p)

**Answer:** .....

- (b) Which of the following are valid variable names i Python? **Mark** one or more choices! (2p)

- A. diameter3                      C. rotation-time                      E. None of the given alternatives are correct  
B. 9gravitation                      D. mass

- (c) What is printed from the following code if the variable `average_temperature` has the value 22? **Mark** your choice! (2p)

```
1 if average_temperature >= 18:
2     print("Tropic", end=' ')
3     print("climate")
```

- A. Tropic                      C. climate                      E. None of the given alternatives are correct  
B. Tropic climate                      D. 22

- (d) Write a single line of code that prints only the first letter from the string variable `answer` to the terminal. (2p)

**Answer:** .....

- (e) What should *condition* be to make the loop in the code below run exactly one thousand times? **Mark** your choice! (2p)

```
1 coins = 0
2 while condition:
3     coins += 1
```

- A. `coins < 1000`                      C. `coins > 1000`                      E. None of the given alternatives are correct  
B. `coins <= 1000`                      D. `coins == 1000`

- (f) Give an example of a number that can be returned from the following call. (You can presume that `random` was imported before the call.) (2p)

```
random.randrange(6)
```

**Answer:** .....

- (g) What of the following code lines can be used to return data from a function? **Mark** one or more choices! (2p)

A. `return variable`

D. `return var1, var2, var3`

B. `# return var1 & var2`

E. None of the given alternatives are correct

C. `import module`

- (h) What does the program do? **Mark** your choice! (2p)

```
1 outfile = open("kalender.txt", "w")
2 for day in calendar:
3     outfile.write(day)
4 outfile.close()
```

- A. Writes elements from the sequence (list) `calendar` to the file `kalender.txt`  
B. Reads days from the file `kalender.txt`  
C. Changes the list `calendar`  
D. Opens the file `kalender.txt` for reading  
E. None of the given alternatives are correct

- (i) What will the content of the list variables be when the following code has run? (3p)

```
1 list1 = [17, 13]
2 list1.append(5)
3 list2 = list1
4 list2.append(21)
5 list3 = []
6 list1.append(8)
7 list3.append(4)
```

`list1:` .....  
`list2:` .....  
`list3:` .....

- (j) What is another word for the `__init__` method in a class? (2p)

**Answer:** .....

## 2. EXPLAIN

$\Sigma$ : 17

- (a) Explain what is the difference between a function definition and a function call? (5p)  
Also give one example of each.

.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....

- (b) What is a parameter? What is an argument? How are they related? (4p)

.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....

- (c) The *dict* data type implements containers of key-value pairs. What are the requirements of *keys* and *values* regarding mutability and uniqueness? (4p)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

- (d) What is the difference between a *function* and a *method*? (4p)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

## 3. READ PYTHON CODE

 $\Sigma$ : 10

(a) What is printed the following code runs?

(4p)

```
1 sum = 0
2 for x in range(2, 8):
3     if x % 2 == 0:
4         for y in range(x):
5             sum = sum + y
6 print('sum:', sum)
```

.....

.....

(b) What is printed when the following code runs?

(6p)

```
1 def banana(size):
2     if size > 100:
3         return "big"
4     else:
5         return "small"
6
7 def pear(size):
8     print("pear", size)
9     grown_size = size + 7
10    return grown_size
11
12 def apple(size):
13     size = pear(size)
14     print("apple", size)
15
16 size = -2
17 print(banana(pear(2)))
```

.....

.....

$\Sigma: 8$

(8p)

(8p)

- (8p)

```
1 tuple_of_tuples = ((1, 2), (3, 4) (5, 9)]
2 tup = Tuple_of_tuples
3 for a b in tup:
4     print('{a} + {b} = {a+b}')
```

This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting practice. There are no margins, text, or other markings on the paper.

## 5. WRITE SOME CODE

 $\Sigma$ : 26

- (a) Write a Python program using a loop that repeats until -1 is entered. The program should ask the user to input an integer number between 0 and 100 and print whether the number is an even number or not. (6p)
- Running the program should look like this.

```
1 Enter a number (0-100): 1
2 Odd number
3
4 Enter a number (0-100): 21
5 Odd number
6
7 Enter a number: 45
8 Odd number
9
10 Enter a number (0-100): 100
11 Even number
12
13 Enter a number (0-100): 57
14 Odd number
15
16 Enter a number (0-100): 100
17 Even
18
19 Enter a number (0-100): -1
20 Goodbye!
```

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....



- (b) The median of the values in a data collection is an alternative to using mean value, especially if the values are very skewed. (10p)

Properties of median:

- For data collections with an odd number of values, the median is the middle value of the sorted collection.
- For data collections with an even number of numbers, the median is the mean value of the two middle values of the sorted collection.

Write a function `median(lst)` that takes a list containing values as a parameter (the list is not empty), and calculates and returns the correct median value.

```
1 >>> median([1, 1, 2, 3, 5, 7, 7, 7, 8])
2 5
3 >>> median([1, 1, 2, 3, 4, 7, 7, 7, 8, 9])
4 5.5
5 >>> median([1, 7, 2, 3, 7, 5, 8, 3, 7, 1])
6 4.0
```

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

- (c) Write a function `to_dict` that takes in a list of words (strings) as a parameter and creates and returns a *dictionary*. (10p)

The keys in the dictionary should be the first letter in each string as an uppercase letter. The value corresponding to the key should be a list of all words from the list beginning with that letter.

**Hint!** The string method `upper()` returns a new string containing the same text as the original string, but in uppercase.

The function head can look like this:

```
1 def to_dict(lst):
2     """
3     Create a dictionary with single uppercase letters used as keys,
4     indexing all words from lst starting with that letter as a list.
5     """
```

If we run the following code with such a function:

```
1 string_list = {'squirrel', 'dog', 'chimpanzee', 'ox', 'lion',
2               'otter', 'seahorse', 'octopus', 'dolphin', 'walrus'}
3 new_dict = to_dict(string_list)
4 print(new_dict)
```

... it will give the following terminal output:

```
1 {'C': ['chimpanzee'],
2   'S': ['squirrel', 'seahorse'],
3   'O': ['octopus', 'ox', 'otter'],
4   'L': ['lion'],
5   'D': ['dolphin', 'dog'],
6   'W': ['walrus']}
```



## 6. CLASSES AND OBJECTS

Σ: 18

- (a) Define a new class **Student** that describes a student with a name and a grade. The constructor should take in two parameters **first** and **last**, but the whole name should be stored as *one string* in *one attribute*. There should also be an attribute **\_grade** with the initial value **None**.

(4p)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

- (b) Write a method `set_grade` that sets a students grade. The grade must be valid though, otherwise nothing should happen. Valid grades are 'A', 'B', 'C', 'D', 'E' or 'F', where 'A' is the highest and 'F' the lowest. In addition a grade can never be lowered. If no parameter value is provided when `set_grade` is called the grade should default to 'E'. Thus you should be able to for example write `stud.set_grade()` or `stud.set_grade('A')` for a `Student` object `stud`. (4p)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

- (c) Write the string representation method for **Student**. The returned string should be on the form **name:grade**. (4p)

Example: 'Carina Nilsson:B'.

If the grade value is missing, dash ('-') should be used as grade character in the string representation.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....



- 
- This image shows a full page of blank, lined paper. It features horizontal ruling lines spaced evenly down the page. Between each pair of ruling lines, there are two rows of small, faint dots, creating a guide for handwriting or note-taking. The paper is otherwise completely blank, with no text or other markings.

