

Άσκηση 1 [3μ]

Απαντήστε στις επόμενες ερωτήσεις πολλαπλής επιλογής:

- (i) Τι από τα ακόλουθα κάνει η συνάρτηση `__init__()` της Python; [1μ]
- (α) Αρχικοποίηση μιας κλάσης προς χρήση.
 - ☒ (β) Η συνάρτηση καλείται όταν δημιουργείται ένα νέο αντικείμενο μιας κλάσης.
 - (γ) Αρχικοποίηση στην τιμή 0 όλων των χαρακτηριστικών δεδομένων (attributes).
 - (δ) Τίποτα από τα προηγούμενα.
- (ii) Ποιο από τα ακόλουθα κάνει χρήση της ιδιότητα της κληρονομικότητας στην Python; [1μ]
- (α) `class Foo: Pass`
 - (β) `class Foo(object): pass` `class Hoo(object): pass`
 - ☒ (γ) `class Foo: pass` `class Hoo(Foo): pass`
 - (δ) Κανένα από τα προηγούμενα.
- (iii) Για το ακόλουθο τμήμα κώδικα, ποια από τις παρακάτω απαντήσεις είναι σωστή; [1μ]
- ```
def printHello():
 print("Hello")
a = printHello()
```
- (α) `printHello()` είναι μια συνάρτηση και η `a` είναι μια μεταβλητή. Καμιά από τις δύο δεν αποτελεί αντικείμενο.
  - ☒ (β) Τόσο η `printHello()` όσο και η `a` αναφέρονται στο ίδιο αντικείμενο.
  - (γ) `printHello()` και `a` αναφέρονται σε δύο διαφορετικά αντικείμενα.
  - (δ) Η ανάθεση συνάρτησης σε μεταβλητή δεν επιτρέπεται στην Python.

**Άσκηση 2 [4μ]**

Να γράψετε το Python πρόγραμμα μιας κλάσης, `IOclass()`, η οποία περιέχει 2 μεθόδους, `passStr()` και `printStr()`. Η μέθοδος `passStr()` δέχεται ένα string από το πληκτρολόγιο, ενώ η `printStr()` τυπώνει την string με κεφαλαία γράμματα.

```
class IOclass():
 def __init__(self):
 self.str1 = ""
 def passStr(self):
 self.str1 = input()
 def printStr(self):
 print(self.str1.upper())

str1=IOclass()
str1.passStr()
str1.printStr()
```

### Άσκηση 3 [4μ]

Να γράψετε μια κλάση *Student* η οποία έχει object attributes τις *age* και *name* ενός φοιτητή και περιέχει δύο μεθόδους, *presentation()* και *print()*. Η πρώτη παρουσιάζει το όνομα του φοιτητή σε format "*Hi! To onoma mou einai:* " και η *print()* παρουσιάζει τα στοιχεία ενός φοιτητή. Θα πρέπει να έχει επίσης μια παράγωγη class *PhDStudent*, με επιπλέον data attribute τον τίτλο του Διδακτορικού του φοιτητή. Η παράγωγη κλάση θα πρέπει να περιέχει μια μέθοδο, *get\_thesis\_title()*, που επιστρέφει τον τίτλο του Διδακτορικού.

```
class Student:
 def __init__(self, name, age):
 self.name = name
 self.age = age
 def presentation(self):
 print 'Hello, my name is ', self.name
 def __str__(self):
 return f'Student: {self.name} is {self.age} years old'

class PhDStudent(Student):
 def __init__(self, name, age, thesis_project):
 self.thesis = thesis_project
 Student.__init__(self, name, age)

 def get_thesis_title(self):
 return self.thesis

stu1=Student('George', 20)
stu=PhDStudent('Fotis',20,'Theory of everything')
stu.presentation()
proj=stu.get_thesis_title()
print(proj)
```

### Άσκηση 4 [4μ]

Έστω ότι δίνεται ένα tuple *t*=(*"cat"*,*"dog"*) και ένα dictionary, *d*={*"day"*:*"Tuesday"*, *"time"*=*"18:00"*). Να γράψετε το πρόγραμμα μιας συνάρτησης *printEverything()* τυπώνει το tuple και το dictionary καθώς επίσης και μια επιπλέον παράμετρο, *period* με default τιμή *'morning'*. Ωστόσο όταν κληθεί η συνάρτηση σύμφωνα με τον χρόνο είναι *'evening'*.

```
def print_everything(period='morning', *args, **kwargs):
 print("It is %s"%period)
 for arg in args:
 print(arg)

 for k, v in kwargs.items():
 print("%s: %s" % (k,v))

Μπορούμε να την καλέσουμε ως εξής:
t = ('cat','dog')
d = {'day':'Tuesday', 'time':'18:00'}
print_everything('evening', *t, **d)
```