E-JUST
الجامعة المصرية اليابانية للعلوم والتكنولوجيا
エジプト日本科学技術大学

CSE 314: Advanced Programming Lab

Lab 4
Object Oriented Concepts

Tuesday 17th November, 2020

Objectives: Review object-oriented concepts.

1 Problem 1

Choose the best answer:

1. Which of the following best describes polymorphism?
   (a) Ability of a class to derive members of another class as a part of its own definition
   (b) Means of bundling instance variables and methods in order to restrict access to certain class members
   (c) Focuses on variables and passing of variables to functions
   (d) Allows for objects of different types and behaviour to be treated as the same general type
2. What is the biggest reason for the use of polymorphism?
   (a) It allows the programmer to think at a more abstract level
   (b) There is less program code to write
   (c) The program will have a more elegant design and will be easier to maintain and update
   (d) Program code takes up less space
3. What is the use of duck typing?
   (a) More restriction on the type values that can be passed to a given method
   (b) No restriction on the type values that can be passed to a given method
   (c) Less restriction on the type values that can be passed to a given method
   (d) Makes the program code smaller

4. What will be the output of the following Python code?

```
class A:
    def __str__(self):
        return '1'
class B(A):
    def __init__(self):
        super().__init__()
class C(B):
    def __init__(self):
        super().__init__()
def main():
    obj1 = B()
    obj2 = A()
    obj3 = C()
    print(obj1, obj2,obj3)
main()
```

(a) 1 1 1

(b) 1 2 3

(c) ‘1’ ‘1’ ‘1’

(d) An exception is thrown

5. What will be the output of the following Python code?

```
class Demo:
    def __init__(self):
        self.x = 1
    def change(self):
        self.x = 10
class Demo_derived(Demo):
    def change(self):
        self.x=self.x+1
        return self.x
def main():
    obj = Demo_derived()
    print(obj.change())

main()
main()
```

(a) 11

(b) 2

(c) 1

(d) An exception is thrown

6. What will be the output of the following Python code?

```
class A:
    def one(self):
        return self.two()
    def two(self):
        return 'A'
class B(A):
    def two(self):
        return 'B'
obj2=B()
print(obj2.two())
```

(a) A

(b) An exception is thrown

(c) A B

(d) B

7. Which of the following statements is true?

   (a) A non-private method in a superclass can be overridden

   (b) A subclass method can be overridden by the superclass

   (c) A private method in a superclass can be overridden

   (d) Overriding isn't possible in Python

2 Problem 2

Part I: Members, Students and Instructors

You're starting your own web development school called Codebar! Everybody at Codebar - whether they are attending workshops or teaching them - is a Member:

- Each member has a full_name.
- Each member should be able to introduce themselves (e.g., "Hi, my name is Kevin!").

Each Member is also either a Student or an Instructor:

- Each Student has a reason for attending Codebar (e.g., "I've always wanted to make websites!").
- Each Instructor a bio (e.g., "I've been coding in Python for 5 years and want to share the love!").
- Each Instructor also has a set of skills (e.g., ["Python", "Javascript", "C++"]).
- An Instructor can gain a new skill using add_skill.

Part II: Workshops

Codebar also has Workshops. Each Workshop has:

- A date.
- A subject.
- A group of instructors.
- A roster of students.
- An add_participant method that accepts a member as an argument. If the Member is an Instructor, add them to the instructors list. If a Member is a Student, add them to the students list.

Create another method print_details that outputs the details of the workshop. Test Your Code

Make your code work for the following calls and print out the response you can see in the comments below:

```
workshop = Workshop("12/03/2014", "Shutl")

jane = Student("Jane Doe", "I am trying to learn programming and need some help")
lena = Student("Lena Smith", "I am really excited about learning to program!")
vicky = Instructor("Vicky Python", "I want to help people learn coding.")
vicky.add_skill("HTML")
vicky.add_skill("JavaScript")
nicole = Instructor("Nicole McMillan", "I have been programming for 5 years in Python and want to spr
nicole.add_skill("Python")

workshop.add_participant(jane)
workshop.add_participant(lena)
workshop.add_participant(vicky)
workshop.add_participant(nicole)
workshop.print_details
# =>
# Workshop - 12/03/2014 - Shutl
#
# Students
# 1. Jane Doe - I am trying to learn programming and need some help
# 2. Lena Smith - I am really excited about learning to program!
#
# Instructors
# 1. Vicky Ruby - HTML, JavaScript
#    I want to help people learn coding.
# 2. Nicole McMillan - Ruby
#    I have been programming for 5 years in Ruby and want to spread the love
#
```