**Lab 15 - Inheritance, Aggregation and Composition**

**Questions:**

**Inheritance**

**1. There are websites such as https://www.carsireland.ie/ that provide information about secondhand vehicles. Design a base class for vehicles with fields such as model year, total mileage, Vehicle Identification Number (VIN), engine, transmission, options, and etc. Design subclasses for car, truck, SUV, and minivan. Think about the specific fields and methods required for the subclasses. Instantiate your classes with examples so you can test your code.**

**2. Given the bankAccount class from Lab 9, question 2, create a subclass MinimumBalanceAccount that inherits bankAccount. MinimumBalanceAccount should have a minimum balance value and overwrite the method withdrawal so the new balance is not below the minimum balance after withdrawing.**

**Composition/Aggregation**

**3. Design a class to represent a Gym. A gym should have members and equipment. Think of what attributes each class should have, and which kind of relationship a member and equipment would have with the gym. Instantiate all your classes and implement a gym with a few equipment and members.**

**4. Complete the code for the classes Student and Registration using composition. Make sure the main code works after adding your implementation.**

|  |
| --- |
| **class Student:**  ***"""***  ***INSERT YOUR DOCSTRING INFORMATION HERE***  ***"""***  **def \_\_init\_\_(self, study\_type, f\_name, l\_name):**  ***# YOUR CODE GOES HERE***  **pass**  ***# YOUR CODE GOES HERE***  **class RegistrationData:**  ***"""***  ***INSERT YOUR DOCSTRING INFORMATION HERE***  ***"""***  **def \_\_init\_\_(self, address, registration\_fee, study\_type, f\_name, l\_name, s\_id="NA"):**  ***# YOUR CODE GOES HERE***  **pass**  ***# YOUR CODE GOES HERE***  ***# MAIN SCOPE - UNCOMMENT IT AND RUN AFTER IMPLEMENTING YOUR SOLUTION***  ***# r = RegistrationData("8 Lower Kevin Street, Dublin 8, Ireland", 1500,***  ***# Student.POSTGRADUATE, "Lucas", "Rizzo")***  ***# r.display\_student\_data()***  ***# print()***  ***# r.set\_student\_id\_property("C12345")***  ***# r.display\_student\_data()***  ***# print()***  ***# for course in ("OOP", "Advanced Databases", "Environmental Analytics"):***  ***# r.get\_student\_object().set\_courses(course)***  ***#***  ***# r.display\_student\_data()***  ***# print()***  ***# print(r.get\_student\_object()) # extra to match the \_\_str\_\_ additional function***  ***# print()***  ***# print(RegistrationData.\_\_doc\_\_)*** |

class Student:

"""

INSERT YOUR DOCSTRING INFORMATION HERE

"""

POSTGRADUATE = 1

UNDERGRADUATE = 2

def \_\_init\_\_(self, study\_type, f\_name, l\_name, courses=None):

self.study\_type = study\_type

self.f\_name = f\_name

self.l\_name = l\_name

if courses == None:

self.courses = []

else:

self.courses = courses

def \_\_str\_\_(self):

output = "First name: " + self.f\_name + "\n"

output += "Last name: " + self.l\_name + "\n"

study\_type = ""

if self.study\_type == Student.POSTGRADUATE:

study\_type = "Postgraduate"

if self.study\_type == Student.UNDERGRADUATE:

study\_type = "Undergraduate"

output += "Study type: " + study\_type + "\n"

if self.courses != None:

count = 1

for course in self.courses:

output += "Course #" + str(count) + ": " + course + "\n"

count += 1

return output

def set\_courses(self, course):

self.courses.append(course)

class RegistrationData:

"""

Stores a student object plus adress, registration fee and id.

Manages student data through get and set methods

"""

def \_\_init\_\_(self, address, registration\_fee, study\_type, f\_name, l\_name, s\_id="NA"):

self.address = address

self.registration\_fee = registration\_fee

self.student = Student(study\_type, f\_name, l\_name)

self.s\_id = s\_id

def display\_student\_data(self):

print(self.student)

print("Address:", self.address)

print("Registration fee:", self.registration\_fee)

print("ID:", self.s\_id)

def set\_student\_id\_property(self, new\_id):

self.s\_id = new\_id

def get\_student\_object(self):

return self.student

# MAIN SCOPE - UNCOMMENT IT AND RUN AFTER IMPLEMENTING YOUR SOLUTION

r = RegistrationData("8 Lower Kevin Street, Dublin 8, Ireland", 1500,

Student.POSTGRADUATE, "Lucas", "Rizzo")

r.display\_student\_data()

print()

r.set\_student\_id\_property("C12345")

r.display\_student\_data()

print()

for course in ("OOP", "Advanced Databases", "Environmental Analytics"):

r.get\_student\_object().set\_courses(course)

r.display\_student\_data()

print()

print(r.get\_student\_object()) # extra to match the \_\_str\_\_ additional function

print()

print(RegistrationData.\_\_doc\_\_)