

BDA200: Foundations of Computer Programming

Assignment 06: Modular Programming, Introduction to Object Oriented Programming

Type: **Individual** | Weight: **15%** | Delivery: A2L – **Assignment submission folder**

Due Date: Please see the **Assignment submission folder or calendar**

Introduction

Object Oriented Programming (OOP) is a design philosophy where code is grouped into collections of related things called objects. The parts that describe an object are generally known as attributes; the parts that describe behaviour are called methods. Objects interface with other objects. A class is a template or definition of an object. And an object is an instance of a class.

As an example, a Person can be a class that has attributes like height, weight, date of birth, language spoken, etc. Methods might include walking, speaking, thinking, eating, etc. An individual like you or I is an object (an instance of a class). We have specific data for height, weight, date of birth, language spoken, etc. We walk, speak, think, eat, etc. in different ways. An object has a reference. In the case of a person, that reference is the name of the individual.

Course Learning Outcomes Evaluated in this Assignment

- Discuss the principles of object-oriented programming
- Demonstrate the use of objects in a computer program

Instructions

You will design a parent class called Vehicle with a number of data attributes and methods that are common to a number of different types of vehicles. The specific vehicles you will be modeling are Car, Truck and Motorcycle.

1. Start by listing data attributes for each of the vehicles (Car, Truck and Motorcycle) on a piece of paper.
2. Those data attributes that are the same should be “pulled out” of the lists and added to a list of attributes for Vehicle
3. Create the Vehicle class with the common data attributes
 - a. Please have at least 3 data attributes
 - b. Please have at least one method that will not be overridden.
 - c. Please have at least one method that will be overridden.
4. Create the three subclasses with the “extra” data attributes.
 - a. Please have at least 1 data attribute that is unique to the class.
 - b. Please have at least 1 method that is unique to the class.
 - c. Please override the method in 3c above.
5. Each of the classes should be in a separate file. These should be imported where appropriate.

6. Create a script that will test the three subclasses. In the tests demonstrate:
 - a. Access to the inherited variables
 - b. Execution of the inherited methods
 - c. Execution of the overridden method(s).
 - d. Execution of the unique method(s).
7. Each module and script should have comments at the top of the code that include (at least) the name of the module/script/class, the author and the date of last update.

All the modules and scripts be zipped into one folder and submitted to the folder in A2L.

For grading information, please see the Rubric that is part of the Assignment information in A2L.