CSCI 6221 – Advanced Software Paradigms
Assignment 2 – Types and OOP concepts
Fall 2024

Due: October 29th, 11:59PM

Submit codes to the GitHub classroom in 3 folders: *task1*, *task2* and *task3*.

1. (30 pts)

Write three functions in C or C++: one that declares a large array statically, one that declares the same large array on the stack, and one that creates the same large array from the heap. Call each of the subprograms many times (at least 100,000) and output the time required by each. Explain the results.

2. (20 pts)

Write a C# program to demonstrate ***dynamic*** and ***static*** binding in OOP. Demonstrate the usage of *virtual*, *new* and *overrid*e keywords.

Note: There is no need to install .net platform – you can use online compilers like www.onlinegdb.com

3. (50 pts)

The task checks your understanding of Object-Oriented Programming (OOP) principles (class design, inheritance, encapsulation, and polymorphism) using Ruby programming language.

You are required to create a library catalog system in Ruby. The system should be able to manage different types of library items, such as books, DVDs, and CDs. Each item should have common properties like *title*, *author*, and *publication year*, but also unique properties based on its type.

Requirements:

1. Create a base class called **LibraryItem** that includes the common properties (e.g., title, author, publication year) and any methods that are shared among all types of library items.
2. Implement three derived classes: **Book**, **DVD**, and **CD**. Each of these classes should inherit from the **LibraryItem** class and include additional properties and methods specific to their type. For example, a Book may have a property like genre, a DVD may have a property like director, and a CD may have a property like artist.
3. Ensure that each derived class has a method called **display_info** that prints out the details of the library item, including both the common properties inherited from the base class and the specific properties of the derived class.

4. Create a class called **Library** that can store and manage a collection of different library items. Include methods for adding items to the library, removing items, and displaying the details of all items in the library.

Demonstrate polymorphism by creating a function in the **Library** class that can display information for any item in the library, regardless of its specific type.

You can test your code with the following guide:

```
# Your implementation goes here

# Create instances of different library items
book = Book.new("Concepts of Programming Languages", "Robert Sebesta", 2015, "Programming")
dvd = DVD.new("Inception", "Christopher Nolan", 2010, "Science Fiction")
cd = CD.new("A Day at the Races", "Queen", 1976, "Rock")

# Display items individually
book.display_info
dvd.display_info
cd.display_info


# Create a library and add items to it
library = Library.new
library.add_item(book)
library.add_item(dvd)
library.add_item(cd)

# Display information about all items in the library
library.display_all_items
```