# Lab 4 – Inheritance, Polymorphism, and Abstract Classes

Maggie Lyon, Nyla Spencer, Ember Roberts

This lab's objective is to examine the concepts of Inheritance, Polymorphism, and Abstract classes. Inheritance is an essential skill in programming that allows for the reusability of code. Polymorphism enhances code flexibility and extensibility, making it easier to accommodate changes and additions to the program. This flexibility and extensibility add to the overall efficiency of the design and implementation process. Abstract classes enable programmers to define common behaviors across multiple related classes while allowing each subclass to provide its own implementation details. Essential skills for building robust and maintainable software solutions include organization and design clarity which is promoted by abstract classes. This lab also employs constructors and overload constructors which are important aspects of object-oriented programming. These allow for encapsulation and abstraction which also promotes cleaner code design.

n the submission, it is essential to outline the accessibility of each member in derived classes, elucidating whether they are inherently unavailable, accessible if not overridden, or possess other specific characteristics based on the inheritance hierarchy. Also, mention what is a virtual function and its purpose?

Upon declaring the StreamingService class, the member variables title and description were set to private, while public getters/setters as well as public functionality functions. This allows the member variables to be protected from unauthorized access in future derived classes. Further, the member functions are accessible in future derived classes, so they can be overwritten on a need basis. One of the member functions, playFunctionality, is declared as a virtual void function. This means that the base class is an abstract class, as virtual functions cannot be used until they are overridden as intended in future derived classes.

All public members of the derived classes are accessible both within the derived class and outside of it, while private members are only accessible within the class itself. If the derived classes were to be declared as an instance of the base class, only the members inherited from the base class would be accessible. The same rules apply for private and oublic memebres as before – private members like title and description are inaccessible, and public members like the getters/setters are accessible.

The following screenshots demonstrate each of the 4 functionalities of the code. These results were exactly as expected.

1. Instance of a movie

```
Press 1 for instance of Movie.
Press 2 for instance of TV show.
Press 3 for instance of Movie declared as a Show.
Press 4 for instance of TV Show declared as a Show.
1
********** Creating an Instance of a Movie **********
Enter Title: The Mario Movie
Enter Description: Animated Movie
Enter Opening Credits: its-a-me!

Details:
Title: The Mario Movie
Descrpition: Animated Movie

Play Functionality:
Opening Credits: its-a-me!
**********
```

2. Instance of a TV show

```
Press 1 for instance of Movie.
Press 2 for instance of TV show.
Press 3 for instance of Movie declared as a Show.
Press 4 for instance of TV Show declared as a Show.
2
********** Creating an Instance of a TV Show **********
Enter Title: The Office
Enter Description: Sitcom
Enter the number of seasons (max 5): 3
Enter the number of episodes per season : 4
Enter runtime of each episode:
S1E1: 20
S1E2: 21
S1E3: 22
S1E4: 23
S2E1: 20
S2E2: 19
S2E3: 20
S2E4: 22
S3E1: 21
S3E2: 20
S3E3: 24
S3E4: 23

Details:
Title: The Office
Descrpition: Sitcom

Play Functionality:
Enter season number: 3
Enter episode number: 1
Run time of selected episode: 21 minutes
**********
```

3. Instance of Movie declared as a Show

```
**************************************************************************
Do you want to continue? (Y/N)y
Press 1 for instance of Movie.
Press 2 for instance of TV show.
Press 3 for instance of Movie declared as a Show.
Press 4 for instance of TV Show declared as a Show.
3
*********Creating an instance of a Movie declared as a Show **********
Enter Title: The Barbie
Enter Description: didnt win an oscar
Enter Opening Credits: greta gerwig

Details:
Title: The Barbie
Descrpition: didnt win an oscar

Play Functionality:
Opening Credits: greta gerwig
**************************************************************************
```

4. Instance of TV show declared as Show

```
*********Creating an instance of a TV Show declared as a Show **********
Enter Title: Gilmore Girls
Enter Description: early 2000's sitcom
Enter the number of seasons (max 5): 2
Enter the number of episodes per season : 2
Enter runtime of each episode:
S1E1: 40
S1E2: 38
S2E1: 42
S2E2: 37

Details:
Title: Gilmore Girls
Descrpition: early 2000's sitcom

Play Functionality:
Enter season number: 2
Enter episode number: 1
Run time of selected episode: 42 minutes
**************************************************************************
```

Contributions:

Nyla – Lab Report

Maggie – Task 1 and 3 and debugging

Ember – Task 2

https://github.com/margaretlyon21/Lab4/tree/main