

DEPARTMENT OF COMPUTER & INFORMATION SCIENCES
CS 995 INTRODUCTION TO PROGRAMMING PRINCIPLES 2018/19
INDIVIDUAL LAB TEST

Duration: 1 hour

Available marks: 100

Contribution to overall mark: This assignment contributes 50% towards your final class mark.

General instructions:

Please read the assignment brief carefully and **attempt all “Assessed Tasks”**. Even if you do not complete everything, make sure that you submit all your code. This is an **open-book** programming lab test (i.e. you are allowed to use the book/lecture notes, your source code from previous practicals/tutorials, etc.), but it is still **under exam conditions** (i.e. no communication among students is allowed). **The use of web browsing will also be heavily monitored.**

Aims:

The aims of this assignment are to:

- implement (in Python) a software system under the paradigm of Object-Orientation; and
- test this system.

Learning outcomes:

After completing this assignment, you will have demonstrated experience of using:

- main control and flow structures of an imperative programming language;
- simple data elements and basic data structures of an imperative programming language;
- the main code structure constructs of an imperative programming language.

[Assignment brief continues on next page]

IMPORTANT - Marking Criteria (breakdown of the 100 available marks):

Your submission will be marked for:

- **Completeness** (i.e. has all required functionality been implemented?), and **correctness** (i.e. does everything work as specified?): As specified by the marks below each ***“Assessed Task”*** – **Total of 80 Marks**
- **Commenting** (i.e. is everything commented as it should?) – **Total 10 Marks**
- **Style** (i.e. code layout, naming conventions, meaningful messages) – **Total 10 Marks**

Submission:

Your lecturer will give you instructions on how to submit your code on MyPlace.

[Assignment brief continues on next page]

Assessed Tasks

Part A: Implementation of classes

1. Implement (in Python) a class named **FilmDirector** that holds the director's name and age. The constructor should set all data field values to meaningful default values. Include methods to set and get the values for each data field.

(5 Marks)

2. Implement (in Python) a class named **DVD** that holds the title, the director (of **FilmDirector** type), and the running time (in minutes). The constructor should:

- i) accept one parameter and use its value to set the value of the director data field, and
- ii) set the values of all other data fields to meaningful default values.

Include methods to set and get the values for each data field. Include a method that prints the details of a **DVD** object. This is an open-ended task, but your method should print the values of ALL data fields along with some meaningful descriptive text.

(10 Marks)

3. Implement (in Python) a class named **MyDVDCollection** that holds a list/array of **DVD** objects and a data field that represents the name of the DVD collection. When a new **MyDVDCollection** object is created, the list/array of **DVD** objects is empty and the data field for the DVD collection name has a meaningful default value. Include methods to get the values for each data field. Include a method that sets the value of the data field for the DVD collection name. Include a method that prints the value of the data field for the DVD collection name, along with the details of ALL **DVD** objects contained in the **MyDVDCollection**.

(15 Mark)

[Assignment brief continues on next page]

4. Implement a method in the **MyDVDCollection** class that takes a **DVD** object as a parameter and works according to the following specification:

If the list **already** contains the parameter **DVD** object, the method should **discard** the parameter and **print** the message ***"This DVD is already in your collection!"*** on the screen. **Otherwise**, the method should add the parameter **DVD** object at the end of the list and **print** the message ***"DVD added successfully to your collection!"*** on the screen.

(10 Marks)

5. Implement a method in the **MyDVDCollection** class that takes no parameters and works according to the following specification:

The method **prints** the message ***"DVD collection is empty!"*** on the screen if the list is empty. Otherwise, it **prints** the message ***"DVD collection is not empty!"*** on the screen.

(10 Marks)

6. Implement a method in the **MyDVDCollection** class that takes a string as a parameter (**the director's name**) and works according to the following specification:

The method **prints** on the screen the details of ALL **DVD** objects directed by the director with the specified name.

(15 Marks)

Part B: Testing

7. Implement a test that **fully** tests the functionality of ***"Assessed Task 4"***.

(5 Marks)

8. Implement a test that **fully** tests the functionality of ***"Assessed Task 5"***.

(5 Marks)

9. Implement a test that **fully** tests the functionality of ***"Assessed Task 6"***.

(5 Marks)

Good luck!!!

[End of assignment brief]