

UNIVERSITY OF WESTMINSTER

COLLEGE OF DESIGN, CREATIVE AND DIGITAL INDUSTRIES

School of Computer Science and Engineering

ONLINE EXAMINATION REFER/DEFER 2019/20

Module Code:	5COSC001W
Module Title:	Object Oriented Programming
Module Leader:	Barbara Villarini
Release Time:	17 July 2020 at 10:00 (Date & Time BST)
Submission Deadline:	17 July 2020 at 14:00 (Date & Time BST)

Instructions to Candidates:

Please read the instructions below before starting the paper

- Module specific information is provided below by the Module Leader
- The Module Leader will be available during the exam release time to respond to any queries via the Discussion Board in the Timed Assessment area of the module's Blackboard site
- As you will have access to resources to complete your assessment any content you use from external source materials will need to be referenced correctly. Whenever you directly quote, paraphrase, summarise, or utilise someone else's ideas or work, you have a responsibility to give due credit to that person. Support can be found at:
<https://www.westminster.ac.uk/current-students/studies/study-skills-and-training/research-skills/referencing-your-work>
- This is an individual piece of work so do not collude with others on your answers as this is an academic offence
- Plagiarism detection software will be in use
- Where the University believes that academic misconduct has taken place the University will investigate the case and apply academic penalties as published in [Section 10 Academic Misconduct regulations](#).
- *Once completed please submit your paper via the Assignment submission. In case of problems with submission, you will have two opportunities to upload your answers and the last uploaded attempt will be marked. Note that instructions on how to compile and submit your handwritten and/or typed solutions will have been sent to you separately.*
- *Work submitted after the deadline will not be marked and will automatically be given a mark of zero*

Module Specific Information

IMPORTANT:

The exam paper is provided in two versions according to the Course you are enrolled on. Answer only the questions in the version corresponding to the programming language that you have been taught during the module:

1) Java Version (pages 2 - 4): BSc Computer Science, BSc Multimedia Computing, BSc Digital Media Development or BEng Software Engineering

2) C++ Version (pages 5 - 6): BSc Computer Games Development

1) JAVA Version

Answer the 8 questions in this section only if you are enrolled on **BSc Computer Science, BSc Multimedia Computing, BSc Digital Media Development or BEng Software Engineering**

DO NOT REPLY to these questions if you are enrolled to BSc Computer Games Development

1) Draw a UML class diagram, of the problem presented below.

A cinema has a name and consists of different screens. Each screen has a name and a number of seats. If the cinema stops to exist, and as a result, it needs to be erased, then the screens should also be erased. Each person working for the cinema has a unique ID and a name and can be a staff member or a projectionist or a cleaner. Each projectionist is responsible of more than one screen.

(12 points)

2) Briefly explain the meaning of the following member visibility operators:

a. +

b. -

c. #

(9 points)

3) Suppose that `class Jazz` and `class Rock` are subclasses of `class Music`. Which of the following are legal?

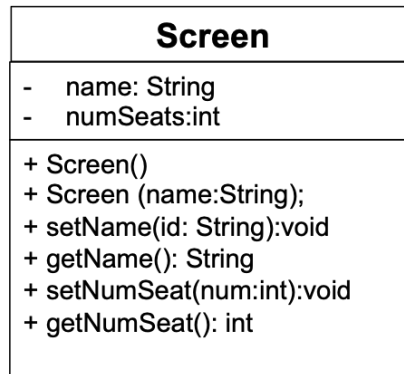
a. `Music m = new Jazz();`

b. `Jazz j = new Rock ();`

c. `Rock s = new Music ();`

(5 points)

4) For all parts of this question, consider the following UML class diagram:



- Provide the implementation for the constructor with no parameters, initialising the instance variables to suitable (valid) start values.
(5 points)
- Provide the implementation for the constructor that takes a String as parameter, which represent the screen name.
(5 points)
- Provide signatures for all the methods in the class.
(5 points)
- Write bodies for the set and get methods for which you provided signatures in question (c) above.
(8 points)
- Write a main method that instantiates the Screen class and use the setter and getter methods you designed.
(5 points)

5) What are the two ways you can provide the implementation for a thread's `run` method and then start the thread? You can explain using an example in Java code.

(10 points)

6) Briefly explain the concept of overriding inherited methods. Use an example to articulate your answer.

(6 points)

7) What are the differences between Swing and AWT?

(6 points)

8) Consider the following class named Movie.

```
public class Movie {  
    private String title;  
    private int duration;  
  
    public Movie(String title){  
        this.title = title;  
    }  
    public void setDuration(int duration){  
        this.duration = duration;  
    }  
    public String getDuration(){  
        return duration;  
    }  
    public String toString(){  
        return "The movie's title is " + title + " and the duration is " +  
duration + "min";  
    }  
}
```

- a. Modify the implementation of the class Movie in order to provide a way to compare different Movies by their duration. Write what you would change in the definition of the class and provide the implementation of the method to be used to compare. Note that you can use from java package `java.lang.Comparable` with the method `public int compareTo(Tobj)`. You don't need to rewrite the whole class, just identifying and rewriting the sections that need changing would suffice.

(10 points)

- b. Write a main method where three Movies are instantiated and stored in an array (or a list) of Movies. For each Movie, set the duration.

(8 points)

- c. Within the main method, sort the array (or list) of Movies according to the criteria implemented in point a. and print the ordered list to the terminal.

(6 points)

END

2) C++ Version

Answer the 8 questions in this section only if you are enrolled on **BSc Computer Game Development**

DO NOT REPLY to these questions if you are enrolled to BSc Computer Science, BSc Multimedia Computing, BSc Digital Media Development or BEng Software Engineering

1) Draw a UML class diagram of the problem presented below. (12 points)

A company has several employees, some of who are managers. A company has at least one division, each of which has employees. Some divisions contain other divisions, and for each division there is one manager who leads that division.

2) Explain the Rule of 3 in your own words. (6 points)

3) Create a `Troop` class. Each troop should have:

- `xPosition` and `yPosition` (both integers) (4 points)
- a suitable constructor (5 points)
- a method `printPosition()` printing its location (5 points)

The class should also:

- keep a list of all the troops so far (6 points)
- have a method `headCount()` returning the size of the list. (6 points)

4) Write some code implementing the following functions: (18 points)

- A function `fill(int n, list<int> &l)` which inserts the numbers 1,...,n into the given list
- a main function which
 - Creates a list `intlist` of integers,
 - launches two threads running `fill(10,intlist)` and `fill(20,intlist)` in parallel,
 - waits for the threads to finish.

You will need to use synchronisation to make sure the threads are never using the list simultaneously.

5) The following questions revolve around implementing a `Vehicle` class.

a) Create an abstract `Vehicle` class with subclasses `Car` and `Bicycle`. Every vehicle has a number of seats for which there should be suitable getter and setter methods. Cars can have between 2 and 6 seats; bicycles between 1 and 2. In order to enforce this, the setter method in each subclass should do nothing, and the constructor should throw an exception, if called with an invalid number. (12 points)

b) Add static `getCar(int n)` and `getBicycle(int n)` methods to the `Vehicle` class which call the relevant constructor. If an exception is thrown, these functions should catch it and print an error message. (6 points)

6) Complete the constructor in the B subclass according to the comment: (8 points)

```
class A{
    int xPos;
public:
    A(int x) : xPos(x) {}
};

class B: public A{
    int yPos;
public:
    // Constructor: sets xPos to x, yPos to y
    B(int x, int y)

};
```

7) Describe the purpose and main parts (classes, relations) of these design patterns:

- a) Observer (6 points)
- b) Factory method (6 points)