A black background with a black square

Description automatically generated with medium confidence

**School of Computing, Engineering and Physical Sciences**

**MSc Information Technology**

**COMP11124 Information Technology Project Management**

**Class Test Student Written Answer Sheet**

**Session 2024/2025 Term 1**

Academic Integrity Statement:

‘I certify that all material in this submission is my own work. I have read and understood the

section in the university regulations regarding collusion, cheating and plagiarism. I confirm

I have not purchased/commissioned material which is presented as my own, including the

use of online services’

Banner ID: B01759301

Name: Prajwalaradhya Shivakumaraswamy Kesaramadu

|  |
| --- |
| (7) Evaluate critically why following the Object-Oriented Paradigm for this library application is more suitable over a procedural approach. (8 Marks) |
| The OOP offers more advantages for this type of application, because of its reusability, abstraction.  Encapsulation.   1. **Encapsulation**: The OOP encapsulates the data (name, available, customer) and behaviour such as *rent\_item*, *return\_item* into reusable and extendable classes, resulting in cleaner code. 2. **Reusability**: Classes such as **Book** and **DVD** inherits from the **RentableObject** abstract class, so reducing the duplication of the logic. 3. **Maintainability**: The abstraction provides base class, so adding new features, modifying the existing functionality is much easier and finding bugs will be easy. 4. **Scalability**: The use of inheritance & polymorphism allows the code to extend to new features & functionality and easy to scale into larger project. 5. OOP design mimics real-world entities such as Books and DVDs, making the code easier to read, develop and compare it to real-world objects. 6. The ***return\_item()*** method is inherited and shared between the child classes reducing unnecessary duplication of code. 7. Each object is self-contained, so it is much easier to develop, test and debug. Allowing only certain parts to debug and test independently without modifying the other objects. 8. The structure of OOP is more easier to read and understand by developers, so it will become easier to understand and whenever if we want to make any new feature it is much more easy to add. |
| (8) If you were to improve this small program, what extra steps would you take (think about SOLID principles or any other OOP topics covered)? (5 Marks) |
| 1. **Single Responsibility Principle**: Using a separate class to handle rental logic, and such as **RentalManager** where it can handle all the renting, returning and handling history details. Then using a **Directory** class to manage the collections of rentable objects like DVD, Book, eBook, Audiobook etc. 2. **Open/Closed Principle**: Making the program for more extendibility, and closed for the modification. Adding an abstract class for rentable items to create more types without modifying the existing code. 3. **Liskov Substitution Principle**: Making all subclasses able to be assigned to **RentableObject** base class. And avoiding subclass-specific code/logic. 4. **Interface Segregation Principle**: Breaking down a large interface/class into smaller, more specific behaviour interface/class. 5. **Dependency Inversion Principle**: Use the dependency injection for calling external dependency such as logging or calling external services/classes. 6. **Saving/Loading**: Enabling the Save and Load functionality to persist the data to the disk using CSV or JSON format, or an external database. |