# Assignment Brief (RQF)

## Higher National Certificate/Diploma in

|  |  |
| --- | --- |
| **Student Name/ID Number:** |  |
| **Unit Number and Title:** | **Unit: 20 Advanced Programming** |
| **Academic Year:** | **2019/2020** |
| **Unit Assessor:** | **Mrs. Vijini Mekala Jayathilake** |
| **Assignment Title:** | **Advanced Programming** |
| **Issue Date:** | **7/7/2020** |
| **Submission Date:** | **29/7/2021** |
| **Internal Verifier Name:** | **Ms Sachini Vindhya** |
| **Date:** | **6/7/2021** |

|  |
| --- |
| **Submission Format:** |
| The submission is in the form of an individual written report. This should be written in a concise, formal business style using single spacing and font size 12. You are required to make use of headings, paragraphs and subsections as appropriate. |
| **Unit Learning Outcomes:** |
| **LO1** Examine the key components related to the object orientated  programming paradigm, analysing design pattern types  **LO2** Design a series of UML class diagrams  **LO3** Implement code applying design patterns  **LO4** Investigate scenarios with respect to design patterns |
| **Assignment Brief and Guidance:** |
| **Scenario**  This scenario is about order management system for United Shop at Baththaramulla. Customer make order. Customers can be either corporate customer or personal customer. Both of them has name and address private type attributes. Corporate customer has contact name and credit limit private attributes. Personal customer has credit card number attribute. That also private. There is a association between employee and corporate customer. Zero or more corporate customer create from employees. An employee can or cannot be corporate customer. order is composed of customer. The order could contain extract one customer, while each customer can make many orders. There is an association between order and order line. Order line has 2 attributes which are private. They are quantity and price. An order can have zero or many order line. Order line and product has association. Research Project Brief  Q1. Assume you work as a junior programmer in Sinha Developers. Currently you are assigned to a development project which is based on Object Oriented Programming. The team leader has assigned you the task to develop a report comprising of the following:   1. Examine the object-oriented programming paradigm characteristics and develop a professionally formatted report that would include these characteristics  * Encapsulation, polymorphism, constructors/destructors, sub objects, abstract/concrete, interface, method, redefinition, generics/templates, containers * Include the information about class relationships like generalization/Inheritance, realization, dependency, aggregation, composition etc. (P1)  1. Determine the design patterns from each of the creational, structural and behavioural pattern types. (M1) 2. Analyse the relationship between OOP paradigm and the design patterns. (D1)   Q2. You are working in Sinha Developers to help improve their documentation for their software development. Your role is to alleviate this situation by developing a series of UML diagrams to reflect the given scenario and to develop new diagrams for future implementation.  You will be required to analyse the existing scenario and then build accurate UML class diagrams. You can make assumption when it is needs. Once completed you will also need to identify possible situations where design patterns would be beneficial and then develop the UML diagrams reflecting the design patterns. (P2, M2, D2)  Q3. You are required to implement design patterns using an IDE such as Eclipse or net beans with Java language. Use an IDE (Eclipse) to develop a small application that implements based on UML class diagram create in Q2 (UML diagrams reflecting the design patterns). (P3, M3, D3)  Q4. There are range of design patterns - creational, structural and behavioral pattern types. Analyze given scenarios(a) to (d) and identify the most appropriate design pattern from a range.   1. You are a member of a programming team, and have been asked to design the class structure for a custom aircraft design system in a computer game. An aircraft will have two basic properties: its name and capacity. Each aircraft can then be customised with various additions. An aircraft may optionally have one or all of a first-class cabin, a viewing area, a communications system and/or a booster rocket. A first-class cabin means the aircraft can have first-class passengers, a viewing area is a place where passengers can watch the scenery below, a communications system is so passengers can talk to people on the ground, and a booster rocket enables the aircraft to move twice as fast. For example, the aircraft GoodCruiser has a capacity of 100 passengers, a first-class cabin and a booster rocket. The aircraft RuthlessRogue has a capacity of 20 passengers, a viewing area and a communications system.   One member of your team quickly begins outlining a set of classes on the whiteboard:  // The generic aircraft  class Aircraft { private int capacity; private String name; }  // An aircraft with an added first-class cabin  class FirstClassCabinAircraft extends Aircraft { }  // An aircraft with an added viewing area  class ViewingAreaAircraft extends Aircraft { }  // An aircraft with an added communications system  class CommunicationsSystemAircraft extends Aircraft { }  // An aircraft with an added first-class cabin and communications system class FirstClassCabinCommunicationsSystemAircraft extends Aircraft { }  What is wrong with the design? Which design pattern would be suitable to remove deficiencies in the design.   1. Consider a scenario of a database server. We want to efficiently manage and restrict the number of connections to databases. Which design pattern would be suitable in this case? 2. Consider a scenario of database transactions. There are different types of database transactions, e.g. add record, update record and delete record etc. If transaction fails, then there is an option of rollback transaction. Which design pattern is suitable to handle these types of transactions and also provides the ability to encapsulate these processes?   (These task achieve P4, M4, D4) |

|  |  |  |
| --- | --- | --- |
| Learning Outcomes and Assessment Criteria | | |
| Pass | Merit | Distinction |
| LO1 Examine the key components related to the object- oriented programming paradigm, analyzing design pattern types | | |
| P1 Examine the characteristics of the object-orientated paradigm as well as the various class relationships | M1 Determine a design pattern from each of the creational, structural and behavioural pattern types | D1 Analyse the relationship between the object-orientated paradigm and design patterns |
| LO2 Design a series of UML class diagrams | | |
| P2 Design and build class diagrams using a UML tool. | M2 Define class diagrams for specific design patterns using a UML tool. | D2 Define/refine class diagrams derived from a given code scenario using a UML tool. |
| LO3 Implement code applying design patterns | | |
| P3 Build an application derived from UML class diagrams. | M3 Develop code that implements a design pattern for a given purpose. | D3 Evaluate the use of design patterns for the given purpose specified in M3. |
| LO4 Investigate scenarios with respect to design patterns code by applying design patterns. | | |
| P4 Discuss a range of design patterns with relevant examples of creational, structural and behavioral pattern types. | M4 Reconcile the most appropriate design pattern from a range with a series of given scenarios. | D4 Critically evaluate a range of design patterns against the range of given scenarios with justification of your choices. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Assignment Feedback** | | | |
| **Formative Feedback: Assessor to Student** | | | |
| **Action Plan** | | | |
| **Summative feedback** | | | |
| **Feedback: Student to Assessor** | | | |
| Assessor Signature |  | Date |  |
| Student Signature |  | Date |  |

**End Of the Assignment**