# Luminus Technical University College - Assignment Brief (RQF)

## Higher National Diploma in computing

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Student Name** | | **Abdelrahamn Saleh** | | | **Language of assessment** | | | **AR** | **EN** |
| **College ID:** | | | **22030961** | |
| **Pearson ID:** | | | **PG76228** | |
| **Unit Number and Title** | |  | **20 Advanced Programming** | | | | | | |
| **Academic Year** | | **2022/2023** | | | | | | | |
| **Unit Tutor** | | **Abdelbaset Assaf, Hazem Al-Najjar** | | | | | | | |
| **Internal Verifier Name and Approval (Signature)** | | **Safa Bani Essa** | | | | **Approval Date: 28/03/20223** | | | |
|  | | | |  | | | |
| **Assignment number and Title** | | **1** | **Software development** | | | | | | |
| **Issue Date (1St Submission)** | | **30/03/2023** | | **Submission Date (1st Submission)** | | |  | | |
| **Issue Date (2nd Submission)** | |  | | **Completion Date (2nd Submission)** | | |  | | |
| **Submission Format** | | | | | | | | | |
| **The submission form is an individual written report, 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, your work must be referenced using Harvard or APA reference style.** | | | | | | | | | |
| **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** | | | | | | | | |
| Transferable skills and competencies developed | | | | | | | | | |
| Understanding how to translate a scenario into UML class diagrams and implementing it using OOP concepts | | | | | | | | | |
| **Vocational scenario:** | | | | | | | | | |
| You are a software developer at Magic solutions. You are tasked with designing a library management system. The system should allow users to search for and borrow books, and it should keep track of which books are available and which are currently checked out.  The library has an admin, collection of books and borrowers. Each book has a title, list of authors, ISBN number and a flag to determine if the book is available or borrowed.  The admin has a salary and working hours. The borrower represents a person which contains the borrower topics interests and a collection of books that are borrowed. Both the admin and the borrower has name and contact information.  Both the library and the borrower should implement a search method to look for a specific book in their collections. Note that the implementation for the search method must be customised and can be different in the two classes.  You should implement a checkout class which contains a method that accepts a book and a borrower as parameters and will check if the book is available. The method should add the book to the borrower and change the book status to borrowed. | | | | | | | | | |
| Assignment activity and guidance | | | | | | | | | |
| **Task 1**  Briefly describe the following Object oriented concepts. Provide diagrams and code snippets (**you must write code**) to supplement your explanations.   1. Access modifiers and encapsulation 2. Composition 3. Inheritance 4. Abstraction 5. Polymorphism   **Task 2**   1. Create a detailed UML class diagram for the **Library Management System** using a suitable UML tool. The classes should include attributes and methods.   **Task 3**   1. **Implement** the system using a suitable object oriented language. Provide **screen shots** (in addition to source code) as evidence of program execution.   **Task 4**   1. What is an example of a creational design pattern, a structural design pattern, and a behavioral design pattern, and how does each pattern address specific challenges in software design and development?   **Task 5**   1. Based on the given code below, draw a UML class diagram. Then explain how you extracted the UML class diagram from the code in details.   public interface Shape {  void draw();  }  abstract class TwoDimensionalShape implements Shape {  private double width;  private double height;    public TwoDimensionalShape(double width, double height) {  this.width = width;  this.height = height;  }    public double getWidth() {  return width;  }    public void setWidth(double width) {  this.width = width;  }    public double getHeight() {  return height;  }    public void setHeight(double height) {  this.height = height;  }  }  class Rectangle extends TwoDimensionalShape {  public Rectangle(double width, double height) {  super(width, height);  }    @Override  public void draw() {  System.out.println("Drawing a rectangle with width " + getWidth() + " and height " + getHeight());  }  }  class Square extends TwoDimensionalShape {  // Constructor  public Square(double width) {  super(width, width);  }    @Override  public void draw() {  System.out.println("Drawing a square with width " + getWidth());  }  }  class Circle implements Shape {  // Fields  private double radius;    public Circle(double radius) {  this.radius = radius;  }    public double getRadius() {  return radius;  }    public void setRadius(double radius) {  this.radius = radius;  }    @Override  public void draw() {  System.out.println("Drawing a circle with radius " + radius);  }  }  class ShapeCollection {  // Fields  private List<Shape> shapes;    public ShapeCollection() {  shapes = new ArrayList<>();  }    public void addShape(Shape shape) {  shapes.add(shape);  }    public void drawAllShapes() {  for (Shape shape : shapes) {  shape.draw();  }  }  } | | | | | | | | | |
| **Recommended Resources**  **Please note that the resources listed are examples for you to use as a starting point in your research – the list is not definitive.**  **Textbooks** | | | | | | | | | |

**Learning Outcomes and Assessment Criteria**

|  |  |  |  |
| --- | --- | --- | --- |
| Pass | Merit | | Distinction |
| **LO1** Examine the key components related to the object orientated programming paradigm, analysing 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.  **D2** Analyse how class  diagrams can be derived from a given code scenario using a  UML tool. |
| **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. | |
| **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. |
|  | | |
|  | |  |