**Kingston University, BSc (Hons) (top-up)**

**Coursework Coversheet**

**Draft Coursework – Subject to Moderation**

**Part 1 - To Remain with the Assignment after Marking**

|  |  |
| --- | --- |
| **Student ID:** | **Student Name:** |
| **Module Code:** | **Module Name:** |
| **Assignment number:** | **ESoft Module Leader:** |
| **Date set:** | **Date due:** |

**Guidelines for the Submission of Coursework**

1. Print this coversheet and securely attach both pages to your assignment. You can help us ensure work is marked more quickly by submitting at the specified location for your module. You are advised to keep a copy of every assignment.

2. Coursework deadlines are strictly enforced by the University.

3. You should not leave the handing in of work until the last minute. Once an assignment has been submitted it cannot be submitted again.

**Academic Misconduct**: **Plagiarism** and/or **collusion** constitute **academic misconduct** under the University's Academic Regulations. Examples of academic misconduct in coursework: making available your work to other students; presenting work produced in collaboration with other students as your own (unless an explicit assessment requirement); submitting work, taken from sources that are not properly referenced, as your own. By printing and submitting this coversheet with your coursework you are confirming that the work is your own.

|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | | ESoft Office Use Only:  Date stamp: work received | | |  | | --- | |  | |

**Kingston University, BSc (Hons) (top-up)**

**Coursework Coversheet**

**Part 2 – Student Feedback**

|  |  |
| --- | --- |
| **Student ID:** | **Student Name:** |
| **Module Code:** | **Module Name:** |
| **Assignment number:** | **ESoft Module Leader:** |
| **Date set:** | **Date due:** |

|  |
| --- |
| Strengths (areas with well-developed answers) |

|  |
| --- |
| Weaknesses (areas with room for improvement) |

|  |
| --- |
| Additional Comments |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **ESoft Module Lecturer:** | **Provisional mark as %:** |  |
| **ESoft Module Marker:** | **Date marked:** |

|  |  |
| --- | --- |
| CI6115 Programming III - Patterns and Algorithms | |
| Weighting | 50% |
| Description | The aim of this coursework is to develop and document a software application using any Object-Oriented Programming (OOP) language and compatible tools of any version. The application can be either console-based or GUI-based and should address the specified case study. |
| Expected Deliverables | The submission must include the following three parts in a single document, along with all project files uploaded as a separate compressed folder:  **Class Diagram and Discussion**:   * A comprehensive class diagram for the proposed system. * Accompanying discussion that highlights the key design decisions and rationale behind them.   **Source Code**:   * Complete source code of the proposed system, demonstrating the use of:   1. Classes and objects   2. Object-Oriented Programming concepts   3. Data structures   4. Relevant algorithms and design patterns   **Test Cases:**   * A full set of test cases used to validate the system |

**Case Study: Smart Library Management System**

You have been assigned the task of designing a **Smart Library Management System (SLMS)** for a university library. The system must support book borrowing, returning, reservations, fine calculation, and notifications for overdue books. It should also allow librarians to manage books, users, and reports efficiently.

### ****Requirements****

1. **Book Management:**
   * Each book has **Book ID, Title, Author, Category, ISBN, Availability Status (Available, Borrowed, Reserved), and Borrowed History**.
   * Librarians can add, update, and remove books.
2. **User Management:**
   * Users have **User ID, Name, Email, Contact Number, Membership Type (Student, Faculty, Guest), and Borrowed Books History**.
   * Membership type affects **borrowing limits and overdue fines**.
3. **Borrowing and Returning:**
   * Users can borrow books if the **Availability Status = Available** and they are within borrowing limits.
   * Borrowed books have a **due date** based on membership type (e.g., 14 days for students, 30 days for faculty).
   * Late returns incur **fines**:
     + Students: LKR 50/day
     + Faculty: LKR 20/day
     + Guests: LKR 100/day
4. **Reservations:**
   * Users can reserve books that are currently borrowed.
   * Once a reserved book is returned, the system should notify the user automatically.
5. **Notifications and Alerts:**
   * Notify users about **due dates, overdue books, and reserved book availability**.
   * Implement **Observer pattern** to manage notifications.
6. **Reports:**
   * Generate reports for librarians, including **most borrowed books, active borrowers, and overdue books**.

**Design Patterns to Apply:**

 **Observer Pattern:** Notify users about overdue books, due dates, and reservation availability.

 **Strategy Pattern:** Implement different fine calculation strategies for Students, Faculty, and Guests.

 **Builder Pattern:** Create complex book objects with optional metadata such as reviews, tags, or editions.

 **Command Pattern:** Represent user actions like borrow, return, reserve, and cancel reservation as commands that can be logged or undone.

 **State Pattern:** Represent book availability states (Available, Borrowed, Reserved).

 **Decorator Pattern:** Add optional features to book objects, such as “Featured,” “Recommended,” or “Special Edition.”

Justify your choice of design patterns based on the specific requirements. Marks will only be awarded if the above criteria comply.

**Further information and reminders:**

* **Class Diagram**: Use any standard UML tool to draw the class diagram. Ensure that the images are clear. Watermarks are acceptable if they are not obtrusive. **Each class name in the diagram must be prefixed with your Kingston University student ID, otherwise, marks will be deducted**.
* **Programming Language**: You may select any object-oriented programming language, but Java is preferable.
* **Frameworks**: Use frameworks with careful evaluation. If a framework generates all the classes for you, marks will be deducted.
* **User Interfaces**: User interfaces can be either Command Line Interfaces (CLI) or Graphical User Interfaces (GUI). Evaluate the user experience in terms of UI/UX for the chosen interface type. The choice of interface type will not impact grading; the focus should be on the quality of implementation.
* **Screenshots**: Include clear screenshots of the code and user interfaces, following the detailed instructions provided.

**Marking Scheme**

|  |  |  |
| --- | --- | --- |
| **Item** | | **Marks Allocation** |
| **Class Diagram (30)** | | |
| 1 | Identification of classes & objects | 15 |
| 2 | Use of associations effectively | 10 |
| 3 | Sensible naming of programmer-defined variables, classes, properties, and methods | 05 |
| **Software Implementation (70)** | | |
| 1 | Completion of functional modules | 25 |
| 2 | Effective use of object-oriented programming concepts | 05 |
| 3 | Effective use of data structures | 05 |
| 4 | Effective use of design patterns | 20 |
| 5 | Justification of the application and design pattern | 10 |
| 6 | Use of Test Cases | 05 |
| **Total** | | **100** |

**Academic Integrity:**

Academic integrity means demonstrating honest, moral behaviours when producing academic work. This involves acknowledging the work of others, giving appropriate credit to others where their ideas are presented as part of your work and the importance of producing work in your own voice. Contributions by artificial intelligence (AI) tools must be properly acknowledged. As part of a learning community students share ideas and develop new ones - you need to be able to interpret and present other people's ideas and combine these with your own when producing work.

**Plagiarism (including copying, self-plagiarism and collusion)**

The act of presenting the work of another person (or people) and/or content generated by artificial intelligence (AI) tools as your own without proper acknowledgement. This includes copying the work of another student or other students.

The University expects students to take responsibility for the security of their work (i.e. with written work, to ensure that other students do not get access to electronic or hard copy of the work). Failure to keep work secure may allow others to cheat and could result in an allegation of academic misconduct for students whose work have been copied, particularly if the origin of the work is in doubt.

**Self-plagiarism**

The act of presenting part or all of your work that has been previously submitted to meet the requirements of a different assessment, except where the nature of the assessment makes this permissible.

**Collusion**

The act, by two or more students of presenting a piece of work jointly without acknowledging the collaboration. This could include permitting or assisting another to present work that has been copied or paraphrased from your own work.

The University also defines collusion as the act of one student presenting a piece of work as their own independent work when the work was undertaken by a group. With group work, where individual members submit parts of the total assignment, each member of a group must take responsibility for checking the legitimacy of the work submitted in his/her name. If even part of the work is found to contain academic misconduct, penalties will normally be imposed on all group members equally.

**Purchasing or Commissioning**

The act of attempting to purchase or purchasing work for an assessment including, for example from the internet, or attempting to commission, or commissioning someone else to complete an assessment on your behalf.

The procedures for investigating suspected cases of academic misconduct are set out in Academic Regulations 6 Academic Integrity - Taught Courses 2023/24

**Acknowledging Generative AI in coursework**

Where generative AI has contributed to an assignment the following information should be included in the submission:

A statement on the use of generative AI as part of the assessment, including the extent of use, and how it was used as part of all stages in creating the final submission, e.g., including planning, and generating ideas. This should normally be provided at the end of a written assignment with the heading ‘Acknowledgement of AI Contribution’. For other assignment types, module staff will advise on how this should be done.

**You must meet all deadlines set. Failure to do so will result in a penalty.**

Work submitted late but within a week of the deadline will be capped at 40% and receive a grade of LP (Late Pass) unless it is not of a passing standard in which case it will receive a grade of LF (Late Fail). Work submitted beyond a week of the deadline without approval will get 0% with a grade of F0. If, however, you have a serious problem, which prevents you from, meeting the deadline you may be able to negotiate an extension in advance. In the first instance you should contact the module team for advice. However, any extension will need to be formally agreed by the Faculty via the Mitigating Circumstances process, your work will then be marked without penalty.