|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **National University of Computer and Emerging Sciences, Lahore Campus** | | | | |
| C:\Users\saif\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.Word\final design.jpg | **Course Name:** | **Software Design & Analysis** | **Course Code:** | **CS324** |
| **Degree Program:** | **BS (CS)** | **Semester:** | **Fall 2021** |
| **Exam Duration:** | **Three hours** | **Total Marks:** |  |
| **Paper Date:** | **30-12-2021** | **Weight** |  |
| **Section:** | **(C &D)** | **Page(s):** |  |
| **Exam Type:** | **Mock Up Final** |  |  |
| No separate answer sheets required; rough sheet are allowed however. Do not attach additional sheets with this question paper. | | | | |

**Question 1 (5+5 marks)**

a) Answer the following multiple-choice questions:

|  |  |
| --- | --- |
| 1. Which of the following is desirable 2. High cohesion and high coupling 3. Low cohesion and low coupling 4. High cohesion and low coupling 5. Low cohesion and high coupling 6. Medium cohesion and medium coupling 7. Solution of the diamond problem in C++ is: 8. Virtual Inheritance 9. Polymorphism 10. Static Binding 11. Dynamic Binding 12. Interfaces 13. Which type of function among the following shows polymorphism?     1. Inline function     2. Virtual function     3. Undefined functions     4. Class member functions     5. None of above | 1. Which of the following pattern is used where we need to treat a group of objects in similar way as a single object? 2. Adapter 3. Composite 4. Singleton 5. Factory method 6. Abstract factory   5. The degree of interaction between different modules is called   1. Cohesion 2. Modularity 3. Maintainability 4. Readability 5. Coupling |

b) Match each entry in the first column with one in the second column:

|  |  |
| --- | --- |
| Class diagram | Support for concurrency |
| Sequence diagram | Transitions change behavior |
| State diagram | Features required in a system |
| Activity diagram | Static structure |
| Use case diagram | Dynamic structure |

**Question 2 (10 marks)**

A bank needs a system to keep track of i) Customers ii) accounts and iii) safe deposit boxes.

Customers will maintain their accounts by making requests to cashiers, who then enter each transaction into a terminal. The cashiers are responsible for accepting or dispensing any money, and provide access to safe deposit boxes.

A customer may have multiple accounts, and may have a single safe deposit box. An account may be a i. share account   ii.  certificate of deposit or ii. a loan account. All accounts generate interest at specific compounding intervals. Besides opening new accounts and closing old ones, customers may check the balance of any open account or request a transaction history. Share account must be either draft accounts, which allow customers to write, cancel, or order checks; or savings accounts, which only permit simple withdrawals.

Certificates of deposit are meant to be redeemed at a specific maturity date, and increase in value until then and they can be cashed out early at their current value, minus a redemption fee. Loans are of a specific amount, and remain open until the entire principal owed is remitted in installments no less than the specified minimum payment. The bank tracks the accumulated interest paid and the outstanding balance, although the customer may be allowed to refinance the loan terms. Some loans are home mortgages, which can have variable interest rates, and must be guaranteed by a mortgage insurer. If the customer defaults, the bank may recoup its losses by repossessing the house for liquidation at its assessed value. The bank also offers college loans, whose terms can depend on various federal programs for which the customer may qualify.

Your task is to draw UML class diagram for above scenario.(There is no need to write data members /functions in class).

**Question 3 (10 marks)**

Consider the following class diagram:



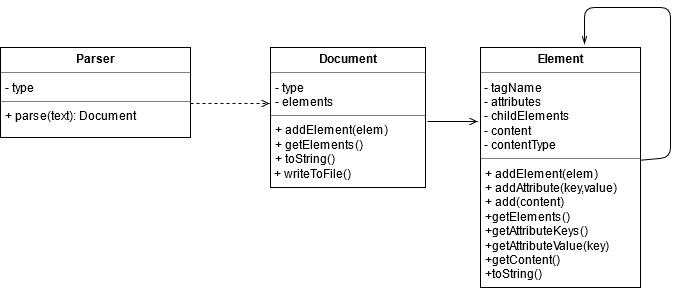
Give sequence diagram showing how total rent for a given customer shall be computed. You may add few simple functions in the classes.

**Question 4 (10 marks)**

|  |  |
| --- | --- |
| <html>  <head></head>  <body>  <p class='text'> Hello world </p>  </body>  </html> | <book category='children'>  <title>Harry Potter</title>  <author>J K. Rowling</author>  <year>2005</year>  <price>29.99</price>  </book> |
| (a) HTML document representing a simple web page | (b) XML document representing a book's information |

HTML and XML are similar markup languages used to structure text: HTML is used for presentation of web content while XML is more general and can be used for representing any information. Both share similar rules and format and their samples are given above, though XML enforces the rules more strictly.

A HTML or XML document comprises of elements (defined by a start and matching end tag that are represented as angular brackets). Each element can contain: attributes (name-value pairs mentioned in the start tag) or body (content enclosed in the start and end tags). Body can be text, comment or further elements.



Consider a proposed UML class diagram for parsing and representing HTML/XML documents. **Parser** parses a given text based upon its type and produces a **Document** instance. **Document** contains 1 or more **Element** instances. XML documents require that one and only one element exists as a root, while there is no such constraint in HTML. A **Document** can be converted to its string (HTML or XML) representation using **toString()** function and so is any **Element**, which shall include content of its children also. Moreover, the **Document** can be written to a file also.

Can you improve the design further with the help of Design Pattern(s)? Show a modified class diagram with proper justification of the Design Pattern(s) used.

**Question 5 (10 marks)**

You are designing an intruder detection system that has following modules (here module refers to one or more classes)

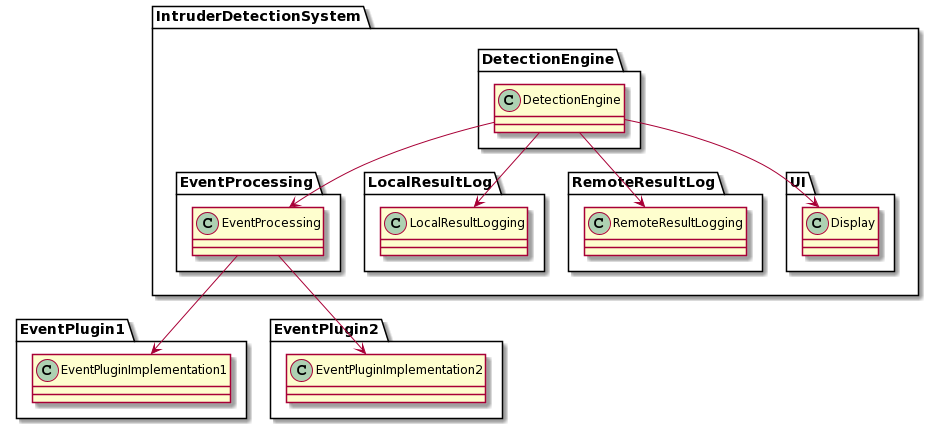
**EventProcessing**: This module processes events coming from EventPlugins and makes them available to the DetectionEngine module for analysis.

**EventPlugins**: These are external components that can be plugged into the system. They provide events which are then processed by the system to detect network intrusion.

**DetectionEngine**: This module implements actual algorithms that detect abnormal activity in the network based on available events.

**ResultLog**: These are two modules that log the detected intrusions locally on hard disk as well as send them remotely to the cloud.

A naive design is the one presented in the UML diagram below.



Your task is to improve this design keeping in mind SOLID design principles. You can add interfaces in any package if required.

After the improvement the design should reflect following characteristics:

1. High level modules should not depend on lower level modules.(hint: DetectionEngine is the highest level module, think about others yourself)
2. Interface with plugins to be designed such that new plugins can be added to the system without a need to make changes to the system
3. Currently there is only two types of logging available, the new design should support adding additional logging modules without impacting the DetectionEngine module
4. Show all relationships (e.g. inheritance, directional association between classes and interfaces)