Object-Oriented Programming (CS F213) Project Questions
IC: Dr. Anita Agrawal **Posted on date: 7/2/25**

Question 1: Smart Home Automation System

Develop a Smart Home Automation System The system should allow users to remotely control and monitor home devices like lights, fans, air conditioning, security systems, and more.
Users should be allowed to turn devices ON/OFF, adjust settings (e.g., temperature, brightness), and schedule automation. It should implement user authentication and role-based access (Admin vs. Regular User). Features like Real-time monitoring of device status, energy consumption should be included. It should support Event-driven automation, such as turning on lights when motion is detected. Ensure secure communication and data handling. Implement a user-friendly GUI or command-line interface.
These basic minimum features are essential, but to establish a comprehensive system, additional functionalities may need to be incorporated.

The entire system has to be designed around OOP using Java.

Minimum 6 classes are required to accommodate all the requirements specified in the design problem. Additionally, it should include the following:

(I) Overloaded methods (minimum 2)
(II) Overloaded constructors (minimum 2)
(III) Vararg overloading (minimum 2)
(IV) Nested classes (static or nonstatic, atleast 1, this is a part of I above)
(V) Abstract class (minimum 1)
(VI) Interface (minimum 1, it can be nested interface or single level or multiple inheritance)
(VII) Hierarchical Inheritance (atleast 1)
(VIII) Multiple Inheritance (atleast 1, this should be in addition to VI above)
(IX) Wrappers
(X) Package
(XI) Exception handling (atleast two cases)
(XII) I/O: File Handling, scanner class etc. (atleast one from each of these)
(XIII) Multithreading (by either Implementing the Runnable interface or extending the thread class)

These are the minimum requirements, but you have the freedom to incorporate a greater number of each as needed.
Note: The names of variables, methods and classes should be lexically rational and should be accompanied with their description in the comments alongside. Your overall code will be something similar to the codes that you wrote in most of your lab hours. It goes without saying that the code should be well indented and should compile and run error free. However, do remember that a non-running code is better than a plagiarized one, and hence the latter one will be penalised heavily if it exceeds 10-15% (may result in recession of this component).

Draw the UML diagram for the above system

Your submission folder should include:

(1) Word doc which explains your project thoroughly along with the UML diagram.
(2) The actual UML diagram
(3) Code appended to the above word doc.
(4) .java files

(5) PPT(you can also show at the time of presentation directly)
(6) Rubrics of the usage of all of the above (I to XIII). Make a table and list the above components and mention how many have you used in each of the category.

Question 2. Personal Finance Management System

Develop a financial tracker that helps users manage their income, expenses, savings, and budgets efficiently. The system should enable users to add income and expenses, categorize spending into various categories such as household expenses, travel, and fuel, and set budgets accordingly. To serve as a comprehensive financial tracker, it should support multiple sources of income. Users should be prompted with options to set budgets on a fortnightly, monthly, or trimester basis.
Key features should include expense tracking, financial summaries, and goal setting. Additionally, the system should incorporate data visualization through graphs and charts to provide valuable insights. To enhance financial discipline, users should receive alerts when expenditures exceed the set budget. Finally, data privacy and ensuring protection through encryption, secure login methods (e.g., OTP, biometric authentication), and cloud backups. Additionally monthly bills, subscriptions, and loan repayments can be automated to avoid missed payments.
Export & Report Generation – Provides options to export financial data in PDF or Excel format for record-keeping.

These basic minimum features are essential, but to establish a comprehensive system, additional functionalities may need to be incorporated.

- The entire system has to be designed around OOP using Java

- Minimum 6 classes are required to accommodate all the requirements specified in the design problem. Additionally, it should include the following:

(I) Overloaded methods (minimum 2)
(II) Overloaded constructors (minimum 2)
(III) Vararg overloading (minimum 2)
(IV) Nested classes (static or nonstatic, atleast 1, this is a part of I above)
(V) Abstract class (minimum 1)
(VI) Interface (minimum 1, it can be nested interface or single level or multiple inheritance)
(VII) Hierarchical Inheritance (atleast 1)
(VIII) Multiple Inheritance (atleast 1, this should be in addition to VI above)
(IX) Wrappers
(X) Package
(XI) Exception handling (atleast two cases)
(XII) I/O: File Handling, scanner class etc. (atleast one from each of these)
(XIII) Multithreading (by either Implementing the Runnable interface or extending the thread class)
These are the minimum requirements, but you have the freedom to incorporate a greater number of each as needed.

Note: The names of variables, methods and classes should be lexically rational and should be accompanied with their description in the comments alongside. Your overall code will be something similar to the codes that you wrote in most of your lab hours. It goes without saying that the code should be well indented and should compile and run error free. However, do remember that a non-running code is better than a plagiarized one, and hence the latter one will be penalised heavily if it exceeds 10-15% (may result in recession of this component).

- Draw the UML diagram for the above system

Your submission folder should include:

(1) Word doc which explains your project thoroughly along with the UML diagram.
(2) The actual UML diagram
(3) Code appended to the above word doc.
(4) .java files
(5) PPT (you can also show at the time of presentation directly)
(6) Rubrics of the usage of all of the above (I to XIII) . Make a table and list the above components and mention how many have you used in each of the category.

Question 3: Smart Parking Management System

Develop an intelligent and automated parking lot management system that helps users find available parking spots efficiently while optimizing space usage. The system should assign parking slots based on vehicle type (car, bike, truck, etc.) and provide a seamless reservation experience. Additionally it should support entry/exit logs for security and analytics. Users should be able to check live parking slot availability and reserve spaces in advance. It should Implement an automated billing system with dynamic pricing (e.g., hourly, daily, or special rates) and support multiple payment methods (credit/debit cards, UPI, e-wallets, etc.). It should track vehicle movement via RFID, QR codes, or license plate recognition to maintain security records. Automated Check-In & Check-Out feature which uses barcodes, NFC, or mobile apps to enable a frictionless parking experience. It can have dynamic pricing model which includes surge pricing during peak hours and special discounts for off-peak periods or long-term parking. It can have Electric Vehicle (EV) Charging Integration which includes dedicated EV charging stations with automated billing for eco-friendly parking solutions. It should support monthly or annual parking subscriptions for regular users with discounted rates.

These basic minimum features are essential, but to establish a comprehensive system, additional functionalities may need to be incorporated.

- The entire system has to be designed around OOP using Java
- Minimum 6 classes are required to accommodate all the requirements specified in the design problem.

Additionally, it should include the following:

(I) Overloaded methods (minimum 2)
(II) Overloaded constructors (minimum 2)
(III) Vararg overloading (minimum 2)
(IV) Nested classes (static or nonstatic, atleast 1, this is a part of I above)
(V) Abstract class (minimum 1)
(VI) Interface (minimum 1, it can be nested interface or single level or multiple inheritance)
(VII) Hierarchical Inheritance (atleast 1)
(VIII) Multiple Inheritance (atleast 1, this should be in addition to VI above)
(IX) Wrappers
(X) Package
(XI) Exception handling (atleast two cases)
(XII) I/O: File Handling, scanner class etc. (atleast one from each of these)
(XIII) Multithreading (by either Implementing the Runnable interface or extending the thread class)

These are the minimum requirements, but you have the freedom to incorporate a greater number of each as needed.

Note: The names of variables, methods and classes should be lexically rational and should be accompanied with their description in the comments alongside. Your overall code will be something similar to the codes that you wrote in most of your lab hours. It goes without saying that the code should be well indented and should compile and run error free. However, do remember that a non-running code is better than a plagiarized one, and hence the latter one will be penalised heavily if it exceeds 10-15% (may result in recession of this component).

- Draw the UML diagram for the above system

Your submission folder should include:

(1) Word doc which explains your project thoroughly along with the UML diagram.
(2) The actual UML diagram
(3) Code appended to the above word doc.
(4) .java files
(5) PPT (you can also show at the time of presentation directly)
(6) Rubrics of the usage of all of the above (I to XIII) . Make a table and list the above components and mention how many have you used in each of the category.

Question 4: Online Food Ordering System

Develop a comprehensive online food ordering platform that enables users to browse restaurants, place orders, and track deliveries seamlessly. The system should support real-time tracking, secure payments, and efficient restaurant management to enhance user experience and operational efficiency. It should include as many features as possible say for customer side, the following:

Users can explore nearby restaurants, view menus, filter by cuisine, pricing, or ratings, and check availability. AI-driven suggestions based on past orders, user preferences, and trending dishes.

Users receive live updates on order status: Order Confirmed → Being Prepared → Out for Delivery → Delivered. Multiple payment options including credit/debit cards, UPI, e-wallets, and cash on delivery (COD). Secure payment. Users can earn rewards, redeem promo codes, and avail of membership benefits. Should show Order History & Reordering – Saves past orders for quick reordering.

Similarly for Restaurants, delivery personnel etc….useful features can be added.

The entire system has to be designed around OOP using Java

- Minimum 6 classes are required to accommodate all the requirements specified in the design problem.

Additionally, it should include the following:

(I) Overloaded methods (minimum 2)
(II) Overloaded constructors (minimum 2)
(III) Vararg overloading (minimum 2)
(IV) Nested classes (static or nonstatic, atleast 1, this is a part of I above)
(V) Abstract class (minimum 1)
(VI) Interface (minimum 1, it can be nested interface or single level or multiple inheritance)
(VII) Hierarchical Inheritance (atleast 1)
(VIII) Multiple Inheritance (atleast 1, this should be in addition to VI above)
(IX) Wrappers
(X) Package

(XI) Exception handling (atleast two cases)
(XII) I/O: File Handling, scanner class etc. (atleast one from each of these)
(XIII) Multithreading (by either Implementing the Runnable interface or extending the thread class)
These are the minimum requirements, but you have the freedom to incorporate a greater number of each as needed.
Note: The names of variables, methods and classes should be lexically rational and should be accompanied with their description in the comments alongside. Your overall code will be something similar to the codes that you wrote in most of your lab hours. It goes without saying that the code should be well indented and should compile and run error free. However, do remember that a non-running code is better than a plagiarized one, and hence the latter one will be penalised heavily if it exceeds 10-15% (may result in recession of this component).

- Draw the UML diagram for the above system

Your submission folder should include:
(1) Word doc which explains your project thoroughly along with the UML diagram.
(2) The actual UML diagram
(3) Code appended to the above word doc.
(4) .java files
(5) PPT (you can also show at the time of presentation directly)
(6) Rubrics of the usage of all of the above (I to XIII) . Make a table and list the above components and mention how many have you used in each of the category.

Question 5: Online Pharmacy and medicine Delivery System

Build an **online medicine ordering system** where users can upload prescriptions, buy medicines, and track orders. It should have the following features: Prescription upload and verification, Automatic reminders for refilling medicines, Integration with teleconsultation services, Secure payment processing and order tracking. Following features can be added from the Doctors' perspective, such as: Users can consult certified doctors online before purchasing certain medications. Doctors can generate digital prescriptions directly on the platform, Health Records Management – Stores user medical history, prescriptions, and past orders securely. It can should have Emergency and SOS features, where it can allow users to request for urgent medicine deliveries.
Security and Compliance: Ensure sensitive health care data to be encrypted.
AI powered System checker: Suggests OTC medicines for common systems.

These basic minimum features are essential, but to establish a comprehensive system, additional functionalities may need to be incorporated.

- The entire system has to be designed around OOP using Java

- Minimum 6 classes are required to accommodate all the requirements specified in the design problem.

Additionally, it should include the following:
(I) Overloaded methods (minimum 2)
(II) Overloaded constructors (minimum 2)
(III) Vararg overloading (minimum 2)
(IV) Nested classes (static or nonstatic, atleast 1, this is a part of I above)
(V) Abstract class (minimum 1)
(VI) Interface (minimum 1, it can be nested interface or single level or multiple inheritance)
(VII) Hierarchical Inheritance (atleast 1)
(VIII) Multiple Inheritance (atleast 1, this should be in addition to VI above)

(IX) Wrappers
(X) Package
(XI) Exception handling (atleast two cases)
(XII) I/O: File Handling, scanner class etc. (atleast one from each of these)
(XIII) Multithreading (by either Implementing the Runnable interface or extending the thread class)

These are the minimum requirements, but you have the freedom to incorporate a greater number of each as needed.

Note: The names of variables, methods and classes should be lexically rational and should be accompanied with their description in the comments alongside. Your overall code will be something similar to the codes that you wrote in most of your lab hours. It goes without saying that the code should be well indented and should compile and run error free. However, do remember that a non-running code is better than a plagiarized one, and hence the latter one will be penalised heavily if it exceeds 10-15% (may result in recession of this component).

- Draw the UML diagram for the above system.

Your submission folder should include:
(1) Word doc which explains your project thoroughly along with the UML diagram.
(2) The actual UML diagram
(3) Code appended to the above word doc.
(4) .java files
(5) PPT (you can also show at the time of presentation directly)
(6) Rubrics of the usage of all of the above (I to XIII) . Make a table and list the above components and mention how many have you used in each of the category.

Question 6: Social media content management

You are tasked with developing a social media content management and analytics platform for influencers, brands, and marketers. Design a system that allows users to schedule posts, track engagement metrics, and analyze audience insights across multiple social media platforms like Facebook, Instagram, Twitter, and LinkedIn.

The system should provide real-time analytics on likes, shares, comments, and follower growth, along with AI-driven recommendations for optimizing post timing and content strategy. Implement features like automated content posting, sentiment analysis for user interactions, hashtag performance tracking, and competitor benchmarking.

Ensure the system supports role-based access (Admin, Content Creator, Marketing Analyst), secure API integration with social platforms, and interactive data visualization (graphs, charts, heatmaps) for trend analysis. Ensure the system supports role-based access (Admin, Content Creator, Marketing Analyst), secure API integration with social platforms, and interactive data visualization (graphs, charts, heatmaps) for trend analysis.

These basic minimum features are essential, but to establish a comprehensive system, additional functionalities may need to be incorporated.

- The entire system has to be designed around OOP using Java

- Minimum 6 classes are required to accommodate all the requirements specified in the design problem.

Additionally, it should include the following:
(I) Overloaded methods (minimum 2)
(II) Overloaded constructors (minimum 2)
(III) Vararg overloading (minimum 2)
(IV) Nested classes (static or nonstatic, atleast 1, this is a part of I above)
(V) Abstract class (minimum 1)
(VI) Interface (minimum 1, it can be nested interface or single level or multiple inheritance)

(VII) Hierarchical Inheritance (atleast 1)
(VIII) Multiple Inheritance (atleast 1, this should be in addition to VI above)
(IX) Wrappers
(X) Package
(XI) Exception handling (atleast two cases)
(XII) I/O: File Handling, scanner class etc. (atleast one from each of these)
(XIII) Multithreading (by either Implementing the Runnable interface or extending the thread class)

These are the minimum requirements, but you have the freedom to incorporate a greater number of each as needed.

Note: The names of variables, methods and classes should be lexically rational and should be accompanied with their description in the comments alongside. Your overall code will be something similar to the codes that you wrote in most of your lab hours. It goes without saying that the code should be well indented and should compile and run error free. However, do remember that a non-running code is better than a plagiarized one, and hence the latter one will be penalised heavily if it exceeds 10-15% (may result in recession of this component).

- Draw the UML diagram for the above system

Your submission folder should include:
(1) Word doc which explains your project thoroughly along with the UML diagram.
(2) The actual UML diagram
(3) Code appended to the above word doc.
(4) .java files
(5) PPT (you can also show at the time of presentation directly)
(6) Rubrics of the usage of all of the above (I to XIII) . Make a table and list the above components and mention how many have you used in each of the category.

Question7: Social media influencer collaboration

You are tasked with developing a social media influencer collaboration and sponsorship platform that connects brands with influencers for marketing campaigns. Design such a system that allows brands to discover influencers, manage campaigns, and track performance metrics across platforms like Instagram, YouTube, TikTok, and Twitter.

The system should enable influencers to create profiles, showcase engagement statistics, and receive sponsorship offers, while brands can filter influencers based on niche, audience demographics, and engagement rates. Implement features like contract management, payment processing, campaign tracking, and AI-driven influencer recommendations.

Ensure the system supports role-based access (Admin, Influencer, Brand Manager, Advertiser), secure authentication, and real-time analytics with data visualization (performance dashboards etc.)

These basic minimum features are essential, but to establish a comprehensive system, additional functionalities may need to be incorporated.

- The entire system has to be designed around OOP using Java

- Minimum 6 classes are required to accommodate all the requirements specified in the design problem.

Additionally, it should include the following:
(I) Overloaded methods (minimum 2)
(II) Overloaded constructors (minimum 2)

(III) Vararg overloading (minimum 2)
(IV) Nested classes (static or nonstatic, atleast 1, this is a part of I above)
(V) Abstract class (minimum 1)
(VI) Interface (minimum 1, it can be nested interface or single level or multiple inheritance)
(VII) Hierarchical Inheritance (atleast 1)
(VIII) Multiple Inheritance (atleast 1, this should be in addition to VI above)
(IX) Wrappers
(X) Package
(XI) Exception handling (atleast two cases)
(XII) I/O: File Handling, scanner class etc. (atleast one from each of these)
(XIII) Multithreading (by either Implementing the Runnable interface or extending the thread class)

These are the minimum requirements, but you have the freedom to incorporate a greater number of each as needed.

Note: The names of variables, methods and classes should be lexically rational and should be accompanied with their description in the comments alongside. Your overall code will be something similar to the codes that you wrote in most of your lab hours. It goes without saying that the code should be well indented and should compile and run error free. However, do remember that a non-running code is better than a plagiarized one, and hence the latter one will be penalised heavily if it exceeds 10-15% (may result in recession of this component).

- Draw the UML diagram for the above system

Your submission folder should include:
(1) Word doc which explains your project thoroughly along with the UML diagram.
(2) The actual UML diagram
(3) Code appended to the above word doc.
(4) .java files
(5) PPT (you can also show at the time of presentation directly)
(6) Rubrics of the usage of all of the above (I to XIII) . Make a table and list the above components and mention how many have you used in each of the category.

Question 8: Smart supply chain and inventory management

You are tasked with developing a smart supply chain and inventory management system for businesses. Design a platform that enables real-time tracking of inventory, automated stock replenishment, and demand forecasting to optimize supply chain operations.

The system should allow suppliers, warehouses, and retailers to manage product stock, track shipments, and monitor sales trends. Implement features like barcode/QR code scanning, AI-driven demand prediction, automated purchase orders, and low-stock alerts.

Ensure the system supports role-based access (Admin, Supplier, Warehouse Manager, Retailer), secure authentication, and data visualization (real-time stock levels, sales trends, and supply chain analytics).

These basic minimum features are essential, but to establish a comprehensive system, additional functionalities may need to be incorporated.

- The entire system has to be designed around OOP using Java

- Minimum 6 classes are required to accommodate all the requirements specified in the design problem. Additionally, it should include the following:

(I) Overloaded methods (minimum 2)
(II) Overloaded constructors (minimum 2)
(III) Vararg overloading (minimum 2)
(IV) Nested classes (static or nonstatic, atleast 1, this is a part of I above)
(V) Abstract class (minimum 1)
(VI) Interface (minimum 1, it can be nested interface or single level or multiple inheritance)
(VII) Hierarchical Inheritance (atleast 1)
(VIII) Multiple Inheritance (atleast 1, this should be in addition to VI above)
(IX) Wrappers
(X) Package
(XI) Exception handling (atleast two cases)
(XII) I/O: File Handling, scanner class etc. (atleast one from each of these)
(XIII) Multithreading (by either Implementing the Runnable interface or extending the thread class)

These are the minimum requirements, but you have the freedom to incorporate a greater number of each as needed.

Note: The names of variables, methods and classes should be lexically rational and should be accompanied with their description in the comments alongside. Your overall code will be something similar to the codes that you wrote in most of your lab hours. It goes without saying that the code should be well indented and should compile and run error free. However, do remember that a non-running code is better than a plagiarized one, and hence the latter one will be penalised heavily if it exceeds 10-15% (may result in recession of this component).

- Draw the UML diagram for the above system

Your submission folder should include:

(1) Word doc which explains your project thoroughly along with the UML diagram.
(2) The actual UML diagram
(3) Code appended to the above word doc.
(4) .java files
(5) PPT (you can also show at the time of presentation directly)
(6) Rubrics of the usage of all of the above (I to XIII) . Make a table and list the above components and mention how many have you used in each of the category.
