Complex Engineering Problem 1

Student Information Management System

In a large university, managing student data efficiently is critical. The Student Information Management System, developed using C programming, is designed to handle various student related operations. The core entities include student details, courses, and grades.

Each student is assigned a unique identifier, name, and relevant course information. Admins can insert, delete, and update student details or course enrollments. The system provides a user-friendly interface for managing student records, including viewing the list of students, course assignments, and grades. The system enables efficient handling of student data, with options to view, update, and search records.

Grades are calculated based on student performance and are recorded in the system for future reference. The system ensures accurate calculations and seamless storage of student data using file handling in C programming. For academic staff, the system provides a way to input and review student performance, making it easy to manage course enrollments and results efficiently. This setup ensures smooth operations within the academic environment, where both students and staff benefit from organized data management, allowing for better student oversight and administration.

Addressed P and K

1. **Depth of Knowledge Required (P1):**

To develop a reliable student management system, developers require an understanding of core software development principles in C programming. In-depth knowledge is necessary in areas like structures, functions, algorithms, and file handling for managing student records and grades. This depth of knowledge is reflected in the following aspects:

- **Engineering Fundamentals (K3):** Developers need to apply structured programming techniques for managing student details, courses, and grades. Implementing features such as inserting, deleting, and updating student records, and calculating grades demonstrates the application of programming fundamentals.
- **Engineering Specialist Knowledge (K4):** This project requires specialized knowledge of academic systems, particularly the management of student data and grade calculation. Developers must understand the specific academic processes and system constraints to tailor the system for seamless student data handling.
- **Engineering Practice (K6):** Practical application involves designing and coding the student management system using C programming. It also includes implementing a

user-friendly interface for academic staff and utilizing file handling to store and retrieve student information efficiently.

2. **Depth of Analysis Required (P3):**

The development of the student Information management system involves analyzing complex academic processes to manage student records, grades, and course enrollments effectively. Developers need to employ analytical thinking to design models for grade calculations, student performance tracking, and academic data storage, ensuring system efficiency and accuracy. Factors such as grading policies, course structures, and performance evaluation metrics must be analyzed to optimize the system's functionality.

3. **Extent of Stakeholder Involvement and Conflicting Requirements (P6):**

The student Information management system involves multiple stakeholders, including university administration, faculty members, and students, each with distinct requirements. Balancing these requirements while maintaining system usability is essential. Developers must collaborate closely with academic staff to ensure the system meets educational needs, from grading to course management, while also addressing students' access to their academic records.

Addressed Engineering Activities (A)

1. Range of Resources (A1):

Use of diverse resources: The development of the student Information management system engages various resources, including people (developers, academic staff, students), money (for development and maintenance), equipment (computers, servers), materials (documentation, user manuals), information (student records, performance data), and technologies (C programming, file handling). These resources collectively contribute to the system's successful creation, implementation, and operation, ensuring efficient management of student data and academic processes.

Complex Engineering Problem 2

Stock and Inventory Management System

In a large retail store, managing stock and inventory efficiently is essential to ensure smooth business operations. The Inventory Management System, developed using C programming, is designed to track stock levels, manage product details, and generate reports. The core entities in the system include products, stock quantities, and transactions.

Each product is assigned a unique identifier, name, category, and price. Admins can insert, delete, and update product details or stock levels. The system provides a user-friendly interface for managing inventory, allowing users to view product lists, check stock availability, and manage orders. The system helps track stock movements, whether goods are added or sold, and updates inventory levels accordingly.

Stock and Inventory reports can be generated to provide insights into stock levels, helping store managers avoid overstocking or stock shortages. The system ensures accurate record-keeping and smooth handling of inventory using file handling in C programming. This setup ensures efficient stock management, enabling the retail store to maintain optimal inventory levels, streamline operations, and improve customer satisfaction.

Addressed P and K

1. Depth of Knowledge Required (P1):

Developing a reliable stock and inventory management system requires a deep understanding of software engineering principles, especially in C programming. In-depth knowledge is necessary in areas such as structures, functions, algorithms, and file handling to manage product data, stock levels, and transactions effectively. The depth of knowledge is demonstrated in the following aspects:

- **Engineering Fundamentals (K3):** Developers need to apply structured programming techniques to manage product details, stock levels, and transactions. Implementing features like adding, deleting, and updating product records and managing inventory updates reflects the application of basic programming concepts.

- **Engineering Specialist Knowledge (K4):** This project requires specialized knowledge of inventory management systems, particularly in tracking stock levels, monitoring product details, and managing transactions. Developers must understand the specific requirements of the retail industry to design a system that handles product information and inventory updates seamlessly.

- **Engineering Practice (K6):** Practical application involves coding the inventory management system using C programming. It also includes designing a user-friendly interface for retail staff and using file handling techniques to store and retrieve product and transaction data efficiently.

2. Depth of Analysis Required (P3):

The development of the stock and inventory management system involves analyzing complex inventory processes, including stock monitoring, product management, and transaction tracking. Developers must apply analytical thinking to design models for managing stock levels, updating inventory, and generating reports. Factors such as restocking strategies, sales trends, and product categorization must be analyzed to optimize the system's functionality and ensure efficient stock control.

3. Extent of Stakeholder Involvement and Conflicting Requirements (P6):

The stock and inventory management system involves multiple stakeholders, such as store managers, sales staff, and customers, each with unique needs. Balancing these needs while maintaining system usability is essential. Developers must work closely with store managers to ensure the system meets operational needs, such as tracking stock levels and generating reports, while ensuring it is easy for sales staff to use when processing transactions.

Addressed Engineering Activities (A)

1. Range of Resources (A1):

Use of diverse resources: The development of the stock and inventory management system engages a variety of resources, including people (developers, store managers, sales staff), money (for system development and maintenance), equipment (computers, servers), materials (user manuals, documentation), information (product data, transaction history), and technologies (C programming, file handling). These resources collectively contribute to the successful creation, implementation, and operation of the system, ensuring efficient management of inventory and product data for smooth retail operations.

Complex Engineering Problem 3

Employee Management System

In a large corporation, managing employee data efficiently is crucial for smooth human resource operations. The Employee Management System, developed using C programming, is designed to handle various employee-related functions such as managing employee records, payroll, and attendance. The core entities in the system include employee details, job roles, salaries, and attendance logs.

Each employee is assigned a unique identifier, along with their name, position, and salary details. Admins can insert, delete, and update employee records or modify their roles. The system provides a user-friendly interface that allows HR personnel to manage employee data, including viewing employee lists, tracking attendance, and calculating salaries. The system also enables payroll processing based on employee attendance and predefined salary structures.

To ensure accurate payroll management, the system calculates employee salaries automatically, factoring in attendance and any bonuses or deductions. This data is stored using file handling techniques in C programming, ensuring secure and efficient record-keeping. This setup ensures streamlined employee management, enabling the HR department to handle employee records and payroll efficiently, benefiting both the corporation and its employees.

Addressed P and K

1. Depth of Knowledge Required (P1):

Developing a reliable employee management system requires in-depth knowledge of software engineering, particularly in C programming. Developers must understand structures, functions, algorithms, and file handling to manage employee records, payroll, and attendance efficiently. The following aspects highlight the required depth of knowledge:

- **Engineering Fundamentals (K3):** Developers must apply structured programming techniques to manage employee details, track attendance, and process payroll. Implementing features such as inserting, updating, and deleting employee records while calculating salaries demonstrates the use of basic programming principles.

- **Engineering Specialist Knowledge (K4):** This project requires specialized knowledge of HR processes, particularly payroll management, attendance tracking, and employee record maintenance. Developers must tailor the system to meet the needs of the HR department by integrating specific business processes, such as salary calculations and attendance monitoring.

- **Engineering Practice (K6):** Practical knowledge is required to design and code the employee management system in C programming. The system also requires a userfriendly interface for HR personnel, with the use of file handling techniques to store and retrieve employee data securely and efficiently.

2. Depth of Analysis Required (P3):

The development of the employee management system involves analyzing complex HR processes, including payroll management, attendance tracking, and employee record maintenance. Developers must apply analytical thinking to design models that handle salary calculations, employee performance tracking, and secure data storage. Factors such as employee roles, salary structures, and attendance policies must be considered to optimize the system's functionality and ensure accuracy.

3. Extent of Stakeholder Involvement and Conflicting Requirements (P6):

The employee management system involves multiple stakeholders, such as HR personnel, employees, and management, each with distinct needs. Developers must balance these needs while maintaining system usability. Collaboration with the HR department is essential to ensure the system meets operational needs, such as managing payroll and tracking attendance, while also ensuring employees can access their records and payroll information.

Addressed Engineering Activities (A)

1. Range of Resources (A1):

Use of diverse resources: The development of the employee management system involves a variety of resources, including people (developers, HR staff, employees), money (for system development and maintenance), equipment (computers, servers), materials (documentation, user manuals), information (employee data, payroll details), and technologies (C programming, file handling). These resources work together to create a functional and efficient system, enabling accurate and streamlined employee management, payroll processing, and attendance tracking.

Complex Engineering Problem 4

Healthcare Management System

In a large healthcare, managing patient data efficiently is crucial for the smooth operation of healthcare services. The Healthcare Management System, developed using C programming, is designed to handle various patient-related operations. The core entities include patient details, doctor assignments, treatments, and medical records.

Each patient is assigned a unique identifier, name, and medical history. Admins and medical staff can insert, delete, and update patient details or treatment information. The system provides a user-friendly interface for managing patient records, including viewing the list of patients, doctor appointments, and treatment histories. The system enables efficient handling of patient data, with options to view, update, and search records.

Medical records are maintained based on patient diagnosis and treatment progress, and they are recorded in the system for future reference. The system ensures accurate and secure storage of patient data using file handling in C programming. For healthcare staff, the system provides a way to input and review patient information, facilitating efficient patient management and care. This setup supports seamless healthcare operations, benefiting both patients and staff by improving data organization, patient tracking, and healthcare administration.

Addressed P and K

1. Depth of Knowledge Required (P1):

To develop a reliable healthcare management system, developers require a deep understanding of core software development principles in C programming. Expertise is needed in areas like structures, functions, algorithms, and file handling to manage patient records and medical information.

- **Engineering Fundamentals (K3):** Developers must apply structured programming techniques for managing patient details, doctor assignments, and treatment records. Implementing features such as inserting, deleting, and updating patient data, and managing medical records demonstrates the application of programming fundamentals.

- **Engineering Specialist Knowledge (K4):** This project requires specialized knowledge of healthcare systems, particularly the management of patient data and medical history. Developers must understand the specific processes in healthcare administration and patient care to tailor the system for efficient and secure data handling.

- **Engineering Practice (K6):** The practical application involves designing and coding the hospital management system using C programming. It also includes implementing a userfriendly interface for healthcare professionals and utilizing file handling for efficient storage and retrieval of patient information.

2. Depth of Analysis Required (P3):

The development of the healthcare management system requires analyzing complex healthcare processes for managing patient records, treatment history, and doctor-patient interactions. Developers must employ analytical thinking to design models for patient care tracking, treatment plans, and medical data storage, ensuring system reliability and accuracy. Factors like healthcare policies, treatment protocols, and patient privacy must be analyzed to optimize system functionality and security.

3. Extent of Stakeholder Involvement and Conflicting Requirements (P6):

The healthcare management system involves multiple stakeholders, including hospital administration, healthcare staff, and patients, each with distinct requirements. Balancing these needs while maintaining system usability and data security is essential. Developers must collaborate with medical professionals to ensure the system meets healthcare needs, from patient tracking to treatment management, while also addressing patient confidentiality and access to medical records.

Addressed Engineering Activities (A)

1. Range of Resources (A1):

Utilizing a wide range of resources: The development of the healthcare management system involves various resources, including people (developers, healthcare staff, patients), money (for development and ongoing maintenance), equipment (computers, medical devices, servers), materials (documentation, user manuals), information (patient medical records, diagnosis, and treatment data), and technologies (C programming, file handling). These resources contribute to the system's successful development, implementation, and operation, ensuring efficient management of patient care and healthcare operations.

Complex Engineering Problem 5

Banking Transaction Management System

In a large financial institution, managing customer accounts and transactions efficiently is essential for operational success. The Banking Transaction Management System, developed using C programming, is designed to handle various account-related operations. The core entities include account details, customer information, transaction history, and banking services.

Each customer is assigned a unique account number, name, and relevant banking details. Admins and bank staff can insert, delete, and update account information, as well as manage customer transactions. The system provides a user-friendly interface for managing account records, including viewing the list of customers, account balances, and transaction histories. The system enables efficient handling of customer data with options to view, update, and search records.

Transactions such as deposits, withdrawals, and fund transfers are recorded in the system for future reference. The system ensures accurate and secure storage of financial data using file handling in C programming. For banking staff, the system provides a way to input and review customer transactions, ensuring accurate financial management. This setup ensures seamless banking operations, improving data management and customer service, while ensuring financial security and regulatory compliance.

Addressed P and K

1. Depth of Knowledge Required (P1):

Developing a reliable banking transaction management system requires an in-depth understanding of core software development principles in C programming. Expertise is necessary in areas such as structures, functions, algorithms, and file handling for managing account details and transaction records.

- **Engineering Fundamentals (K3):** Developers must apply structured programming techniques for managing account details, customer transactions, and banking services. Implementing features such as inserting, deleting, and updating account data, and performing accurate financial calculations demonstrates the application of programming fundamentals.

- **Engineering Specialist Knowledge (K4):** This project requires specialized knowledge of banking systems, particularly in the management of customer data and financial transactions. Developers must understand the banking industry's specific operational and regulatory processes to tailor the system for secure and efficient data handling.

- **Engineering Practice (K6):** Practical application involves designing and coding the bank account management system using C programming. It also includes implementing a userfriendly interface for banking staff and using file handling techniques to securely store and retrieve customer information and transaction records.

2. Depth of Analysis Required (P3):

The development of the banking transaction management system requires analyzing complex financial processes for managing customer accounts, transaction histories, and regulatory compliance. Developers must use analytical thinking to design models for tracking account balances, performing financial transactions, and securing sensitive banking data. Factors such as transaction rules, regulatory requirements, and security protocols must be analyzed to optimize system performance and security.

3. Extent of Stakeholder Involvement and Conflicting Requirements (P6):

The banking transaction management system involves multiple stakeholders, including bank management, employees, regulators, and customers, each with different requirements. Developers must balance these needs while ensuring the system remains secure, user-friendly, and compliant with financial regulations. Close collaboration with banking professionals is required to ensure the system meets operational needs, manages customer data accurately, and adheres to legal and security standards.

Addressed Engineering Activities (A)

1. Range of Resources (A1):

Utilizing a diverse range of resources: The development of the banking transaction management system involves various resources, including people (developers, bank staff, customers), money (for system development, upgrades, and maintenance), equipment (computers, servers, ATMs), materials (documentation, user manuals), information (customer data, transaction records, banking rules), and technologies (C programming, file handling, security protocols). These resources contribute to the system's successful design, development, implementation, and operation, ensuring efficient financial management and customer service.

Complex Engineering Problem 6

Library Management System

In a university setting, efficient management of book circulation, user information, and inventory tracking is crucial to maintaining an effective learning environment. The Library Management System, developed using C programming, is designed to automate various library operations, including book lending, user registration, return processing, and inventory updates. Key entities in this system include books, users, borrowing records, and stock status.

Each book is uniquely identified with attributes such as ID, title, author, subject, and availability status. Administrative users can insert, delete, or update book information, while students and faculty can search for books by title or subject, check their availability, and initiate borrow requests. The system logs borrowing and return actions and automatically updates book availability accordingly. Borrowing durations and penalties for late returns are calculated and recorded.

Data security and integrity are maintained using file handling mechanisms in C programming. The system ensures operational efficiency, minimizes manual record-keeping errors, and enhances the experience of both library staff and users through streamlined access and management.

Addressed P and K

1. Depth of Knowledge Required (P1):
Creating a fully functional library management system requires a strong understanding of software development fundamentals, specifically in C programming. Developers must apply knowledge of structures, arrays, functions, and file operations to build a system capable of managing books, users, and borrowing transactions.

- **Engineering Fundamentals (K3):**
  Developers employ structured programming techniques to build logic for book cataloging, inventory tracking, borrowing and return operations, and penalty calculations, showcasing core programming skills.

- **Engineering Specialist Knowledge (K4):**
  The project involves domain-specific knowledge of library operations such as managing lending periods, overdue fines, and catalog classifications to design a system that reflects real library workflows.

- **Engineering Practice (K6):**
  The practical aspect involves implementing a user-friendly interface with persistent storage, using file handling in C to manage book and user data. Real-world constraints such as late return penalties and concurrent borrowing are also addressed.

2. Depth of Analysis Required (P3):
The system development requires in-depth analysis of borrowing workflows, overdue return management, and inventory synchronization. Developers must account for time-based rules, resolve data consistency issues, and ensure accurate penalty assessments. Abstract modeling and original problem-solving are essential to meet operational demands and avoid system conflicts.

3. Extent of Stakeholder Involvement and Conflicting Requirements (P6):
The system addresses the needs of diverse stakeholders including librarians, students, faculty members, and administrative staff. Librarians prioritize accurate inventory management and overdue tracking, while users demand fast and efficient book searching and borrowing features. Reconciling these varying expectations requires stakeholder engagement and thoughtful system design that balances usability with operational rigor.

Addressed Engineering Activities (A)

1. Range of Resources (A1):
The development and implementation of the Library Management System require a diverse set of resources to ensure effective functionality and scalability. Human resources include developers for coding and maintenance, librarians who operate the system, and users such as students and faculty who interact with it daily. Financial resources are necessary for software development, infrastructure setup, and ongoing system support. Technological infrastructure such as library computers, terminals for users, and networking equipment enables system accessibility and performance. In addition, essential materials such as printed reports, user guides, and help documentation support the usability and training aspects of the system. Information resources, including book metadata, user registration data, and borrowing logs, form the core dataset for the system’s operation. Technologically, the system leverages the C programming language with file handling techniques and a terminal-based interface to manage data efficiently and securely. These combined resources contribute to the system's effectiveness in automating library operations and enhancing user experience.

Complex Engineering Problem 7

Hotel Reservation Management System

Managing room bookings, guest records, and billing in a high-demand hospitality environment requires a reliable and automated solution. The Hotel Reservation Management System, developed using C programming, streamlines key hotel operations such as room reservations, cancellations, occupancy tracking, guest data management, and invoice generation. The system is composed of modules for room inventory, customer interaction, dynamic pricing, and secure billing processes.

Each hotel room is uniquely identified and associated with its type (e.g., single, double, suite), nightly rate, and availability status. Administrative users can update room details, monitor occupancy levels, and apply seasonal or promotional pricing. Guests can interact with the system to search for available rooms, place bookings, and cancel reservations when needed. During checkout, the system computes total charges based on stay duration, applies taxes and discounts, and generates an invoice. All booking and guest data are securely stored using file handling techniques in C to ensure data integrity and accessibility.

The system enhances hotel operational efficiency by minimizing booking conflicts, automating billing, and improving customer service through a structured reservation workflow.

Addressed P and K

1. Depth of Knowledge Required (P1):
Developing this hotel reservation management system demands a comprehensive understanding of software development principles in C, especially for implementing date-based logic, managing reservation records, and handling billing transactions.

- **Engineering Fundamentals (K3):**
  Structured programming techniques are applied to build modules that manage room information, booking timelines, availability tracking, and guest services while maintaining data accuracy and integrity.

- **Engineering Specialist Knowledge (K4):**
  This project incorporates domain-specific knowledge of hotel operations, including dynamic room pricing, tax application, invoice generation, and customer service protocols.

- **Engineering Practice (K6):**
  Real-world software engineering practice is demonstrated through the development of file-based storage mechanisms, input validation processes, and report generation functionalities that support hotel staff in daily operations.

2. Depth of Analysis Required (P3):
Building the reservation system requires in-depth analysis of real-life hospitality scenarios such as overlapping bookings, high-demand seasons, and cancellation policies. Developers must design logical models for allocating rooms, tracking guest preferences, and generating accurate bills based on variable pricing and taxes, ensuring reliable and optimized service delivery.

3. Extent of Stakeholder Involvement and Conflicting Requirements (P6):
Multiple stakeholders contribute to system requirements: hotel staff need control over room inventory and billing; guests demand a seamless and intuitive reservation experience; and managers require accurate financial reports. These stakeholder needs may conflict at times—such as balancing system complexity with ease of use—requiring careful planning and a user-centered design approach to meet operational and service expectations.

Addressed Engineering Activities (A)

1. Range of Resources (A1):
The development and implementation of the Hotel Reservation Management System rely on a diverse set of resources. Human resources include developers who create the system, hotel staff such as front desk personnel and accountants who operate it, and guests who interact with it. Financial resources are required for development, deployment, and regular updates. Technological infrastructure such as desktop computers, reservation terminals, and receipt printers is essential for operation. Supporting materials like reservation logs, printed receipts, and tax documentation assist in record-keeping and audits. Information resources including guest records, room availability status, seasonal pricing strategies, and applicable tax regulations are integral to the system's functioning. Technologically, the system utilizes C programming, file handling, and a command-line or graphical user interface for interaction. These combined resources ensure the solution is robust, scalable, and tailored to meet the daily needs of hotel management and guests.

Complex Engineering Problem 8

Transport Booking System

For a transportation company, ensuring efficient ticket booking and vehicle tracking is vital to improving logistics, reducing operational errors, and enhancing customer satisfaction. The Transport Booking System, developed using C programming, is designed to automate key operational components such as vehicle management, route scheduling, real-time seat allocation, and fare calculation. The system integrates both administrative and user-facing functionalities to streamline transportation services.

Administrators can configure new vehicles, define and update routes, manage trip schedules, modify seat availability, and cancel trips when required. Passengers can browse available routes, check departure times, and book tickets based on current seat availability. Fare calculation is automated, based on distance and route selection. All booking and cancellation transactions are logged, and the system dynamically updates seat occupancy. Data security and accuracy are ensured using file handling in C, maintaining consistent records of passengers, routes, and transactions.

The system minimizes manual errors, prevents overbooking, and supports real-time updates for route management and reporting, making it a powerful tool for transportation companies seeking to optimize their services and improve the overall passenger experience.

Addressed P and K

1. Depth of Knowledge Required (P1):
Creating a robust transport booking system requires advanced knowledge of C programming along with an understanding of transportation operations. Developers must implement logical data relationships between routes, schedules, vehicles, and customer actions.

- **Engineering Fundamentals (K3):**
  Structured programming techniques are used to manage route creation, vehicle registration, schedule updates, and seat tracking in real-time, requiring effective data modeling and control flow management.

- **Engineering Specialist Knowledge (K4):**
  Developers must be familiar with fare structures, distance-based pricing models, and route optimization to translate complex transport concepts into efficient algorithms implemented in C.

- **Engineering Practice (K6):**
  Practical implementation involves file-based data storage, user and admin interface design, validation logic, and reporting features, ensuring consistent service delivery across different user roles.

2. Depth of Analysis Required (P3):
Developing the system involves analyzing how various components—routes, times, vehicles, and bookings—interact in real-time. Developers must design the system to handle peak hours, sudden cancellations, and conflicting schedules, while supporting scalable pricing models and maintaining data consistency across all modules.

3. Extent of Stakeholder Involvement and Conflicting Requirements (P6):
Multiple stakeholders are involved: passengers need quick booking confirmations and route visibility; drivers require accurate assignments and trip details; management seeks real-time reports on fleet performance and revenue. Balancing the usability expectations of passengers with operational control for staff requires detailed planning and clear stakeholder communication.

Addressed Engineering Activities (A)

1. Range of Resources (A1):
The development and operation of the Transport Booking System rely on a broad spectrum of resources. Human involvement includes developers who build the system, booking agents who operate it, drivers who depend on schedules and route assignments, and passengers who interact with it for reservations. Financial investment is required for system development, maintenance, and infrastructure such as ticketing terminals. Equipment like booking kiosks, ticket printers, and in-vehicle terminals are essential for deployment. Supportive materials including printed route maps, travel schedules, and documentation help guide both users and staff. Key informational resources include vehicle records, route details, seat inventory, and fare policies. Technologically, the system is built on C programming principles using file systems, terminal-based interfaces, and server logs for monitoring. Together, these resources enable reliable, scalable, and responsive transportation services powered by automation.

Complex Engineering Problem 9

Event Management System

Coordinating events in educational institutions or community settings demands a structured approach to scheduling, registration, and feedback collection. The Event Management System, developed using C programming, addresses these needs by automating the creation and management of events, tracking participant registrations, monitoring schedules, and generating reports. It enhances the overall efficiency of event coordination by reducing dependency on manual tracking and improving the accuracy of data handling.

Each event is uniquely defined by its title, description, date, time, and venue. Event organizers can add, update, or delete events, monitor participant registration, manage attendance, and produce summary reports. Participants interact with the system to register for events and receive confirmation, ensuring smooth communication and data consistency. The system helps identify scheduling conflicts and supports effective venue allocation. All records—ranging from event details to participant feedback—are stored securely using file handling in C programming, ensuring long-term accessibility and data integrity.

By automating major event processes and centralizing record-keeping, the system minimizes logistical complexity, supports real-time updates, and improves stakeholder coordination.

Addressed P and K

1. Depth of Knowledge Required (P1):
Developing this system requires a strong command of C programming concepts, particularly those used for modeling structured data, managing user inputs, and organizing time-based activities. Additionally, an understanding of event logistics is necessary to implement practical scheduling and registration solutions.

- **Engineering Fundamentals (K3):**
  Structured programming techniques are applied to manage event data, register participants, and track attendance efficiently. Arrays and structures are used to organize event-related information logically and consistently.

- **Engineering Specialist Knowledge (K4):**
  Domain-specific knowledge of event management—such as scheduling protocols, venue coordination, and participant flow—is essential to build a system that mirrors real-world planning challenges.

- **Engineering Practice (K6):**
  The use of file handling to store event details, participant data, and feedback reflects

practical engineering needs. The design of the user interface for both organizers and participants emphasizes functionality, data accuracy, and user experience.

2. Depth of Analysis Required (P3):
Designing the system requires analysis of scheduling constraints, participant load, and venue limitations. Developers must model how multiple events interact over time and space, avoid conflicts, and allow for flexible event coordination. Analytical thinking is essential to manage time overlaps, track capacity limits, and support concurrent workflows between various system users.

3. Extent of Stakeholder Involvement and Conflicting Requirements (P6):
The Event Management System supports various stakeholders—organizers, volunteers, and participants—each with distinct roles and expectations. Organizers need control over event logistics, volunteers require clarity on schedules and responsibilities, and participants expect a seamless registration process. Resolving these potentially conflicting needs demands thoughtful interface design and collaborative stakeholder input during development.

Addressed Engineering Activities (A)

1. Range of Resources (A1):
The development and deployment of the Event Management System involve a comprehensive mix of resources. Human contributors include event organizers who manage schedules, volunteers assisting in execution, participants interacting with the system, and developers who build and maintain the platform. Financial resources are allocated for technical setup, system maintenance, promotional materials, and event-related logistics. Equipment such as registration booths, input terminals, and feedback kiosks support on-ground operations. Materials like flyers, printed schedules, and registration forms aid in both digital and physical communication. Core informational resources—event details, time slots, participant records, and feedback—form the backbone of the system. Technologically, the solution utilizes C programming with file handling and terminal-based interfaces to provide a reliable and user-friendly experience. Together, these resources facilitate streamlined event management and improved stakeholder satisfaction.

Complex Engineering Problem 10

Online Examination System

In academic institutions, administering examinations that are secure, scalable, and efficient is a persistent challenge. The Online Examination System, developed using C programming, offers a robust solution by automating critical exam functions such as question creation, user authentication, timed assessments, and automated result processing. The system serves both administrative users and students, ensuring smooth, reliable, and transparent execution of digital tests.

Administrative users can create and configure multiple-choice or short-answer quizzes, assign marks per question, and define exam duration. Students can securely log in to access scheduled exams, where the system manages question display, enforces time constraints, and captures answer submissions. After submission, the system automatically evaluates responses and either displays immediate feedback or securely stores results for later analysis. All operations—question storage, user credentials, exam logs, and grading—are handled using C file management techniques, ensuring data consistency and integrity.

This digital approach minimizes the need for paper-based assessments, reduces manual marking errors, and supports academic integrity through secure login, random question ordering, and controlled access to test content. It enhances the fairness, efficiency, and scalability of examinations across various academic programs.

Addressed P and K

1. Depth of Knowledge Required (P1):
Developing a reliable online examination platform necessitates solid expertise in C programming and an understanding of secure logic flows and academic assessment principles.

- **Engineering Fundamentals (K3):**
  The system utilizes structured programming techniques to handle login validation, question navigation, scoring logic, and secure session management. Arrays, structures, and control flow are essential to simulate real-time testing environments.

- **Engineering Specialist Knowledge (K4):**
  A strong grasp of academic examination processes—including grading policies, question randomization, and secure data access—is required to design a fair and tamper-proof system. Developers must ensure mechanisms are in place to uphold exam integrity.

- **Engineering Practice (K6):**
  Practical implementation includes creating a timed quiz interface, managing real-time

user input, and storing exam content and results securely via file I/O. The system mirrors the core elements of modern online testing platforms at a foundational level.

2. Depth of Analysis Required (P3):
The design process involves analyzing numerous exam-related parameters—such as question sequencing logic, session timeouts, input validation, and scoring algorithms. Developers must account for incomplete submissions, answer conflicts, and timing errors, requiring thoughtful data modeling and flow control.

3. Extent of Stakeholder Involvement and Conflicting Requirements (P6):
Stakeholders include teachers, students, and institutional administrators. Faculty require tools for secure question management and accurate grading; students demand fairness, clarity, and accessibility; administrators seek system reliability and reporting features. Meeting all these needs involves resolving conflicting priorities through user-centric and technically sound design.

Addressed Engineering Activities (A)

1. Range of Resources (A1):
The development and deployment of the Online Examination System engage a variety of resources to ensure successful implementation and use. Human resources include developers for coding and system design, instructors for question input and grading configurations, students as system users, and IT administrators for technical support and system supervision. Financial resources are required for system development, future scaling, and maintenance. Technical infrastructure includes computers, local or server-based exam terminals, and potentially network access for future upgrades. Materials such as exam policies, question banks, and result forms provide content and compliance documentation. Critical informational assets include student profiles, test content, and performance metrics. Technologically, the system is built using C programming and incorporates file handling, terminal interfaces, and secure input/output management to support a fair, efficient, and reliable digital examination experience.

Complex Engineering Problem 11

Garage Service Management System

A busy auto repair garage requires an efficient, automated system to manage vehicle intake, repair workflows, spare parts tracking, and customer billing. The Garage Service Management System, developed using C programming, streamlines garage operations by facilitating structured service logging, diagnostics, billing, and record maintenance. The system improves operational efficiency, reduces manual error, and enhances customer experience through transparent and traceable job tracking.

Each incoming vehicle is logged with a unique ID, along with the owner's personal details, reported issues, current service status, and repair charges. Garage staff, including administrators and technicians, can update the repair status at various stages—such as inspection, diagnosis, and completion—and add notes for each phase. The system tracks the parts used and calculates charges for labor and materials. A detailed invoice is automatically generated upon service completion. Additionally, service history is maintained to support diagnostics in recurring visits, contributing to a personalized and professional customer experience.

Using file handling techniques in C, the system ensures secure and accurate storage of all critical data, including customer profiles, vehicle service records, billing information, and part usage logs.

Addressed P and K

1. Depth of Knowledge Required (P1):
The development of this system requires deep understanding of structured programming in C and the ability to model domain-specific operations based on real-world garage workflows.

- **Engineering Fundamentals (K3):**
  Core programming techniques are applied to handle customer registration, vehicle servicing status, charge computation, and spare parts management. Control structures, data arrays, and structures are used to track job flows from vehicle intake to billing.
- **Engineering Specialist Knowledge (K4):**
  Knowledge of automotive service processes—such as labor hour tracking, spare parts management, service timeline estimation, and job card generation—is essential for modeling garage operations accurately.
- **Engineering Practice (K6):**
  Developers implement file handling for maintaining persistent records of services, repairs, and invoices. A user-friendly interface for mechanics and staff supports real-time data entry and retrieval for smooth daily operations.

2. Depth of Analysis Required (P3):
System development requires modeling interconnected data such as repeat service patterns, estimated repair durations, and inventory movement. Developers must build logic for managing concurrent service jobs, preventing part shortages, and estimating job timelines. Analytical thinking supports building insights into customer behavior and vehicle repair trends.

3. Extent of Stakeholder Involvement and Conflicting Requirements (P6):
Stakeholders include customers who expect prompt and transparent service; mechanics who need streamlined job tracking and part availability; and inventory managers who aim to control costs and minimize excess stock. Balancing these priorities requires a system design that addresses real-time service needs while maintaining efficient resource management and reporting clarity.

Addressed Engineering Activities (A)

1. Range of Resources (A1):
The development and functioning of the Garage Service Management System rely on a diverse set of resources. Human resources include garage staff, technicians, customers, and system operators responsible for service coordination and interaction with the system. Financial resources are involved in budgeting for spare parts, labor costs, and billing operations. Technical infrastructure such as computers, diagnostic equipment, and printers supports real-time service entry and invoice generation. Supporting materials like service checklists, job cards, and printed invoices help ensure procedural accuracy. Critical data such as vehicle history, customer profiles, service logs, and parts usage records are used to inform operations. Technologically, the system is built using C programming, incorporating file handling techniques and logic for interactive billing and repair tracking. These resources collectively ensure that the garage operates with improved precision, service quality, and customer satisfaction.

Complex Engineering Problem 12

Medical Store Management System

To modernize the operations of a local pharmacy, the Medical Store Management System is developed using C programming, providing a digital solution to streamline inventory control, medicine management, and customer interaction. This system consists of two primary components: an administrative panel for internal pharmacy staff and a customer-facing interface for external users.

Through the admin panel, authorized users can add, delete, or update medicine information such as name, price, and quantity. Staff can search for specific medications and monitor inventory levels in real time. The customer interface allows users to browse available medicines, search for specific items, and add products to their virtual cart. The system automatically calculates the total cost including taxes and enables secure payment processing. Upon successful payment, inventory levels are updated to reflect the transaction. All medicine and customer data are stored securely using file handling in C programming, ensuring data privacy and consistency.

The implementation of this system improves inventory visibility, reduces manual errors, streamlines sales processes, and delivers an enhanced customer experience. It also supports operational accuracy and compliance in the pharmacy domain.

Addressed P and K

1. Depth of Knowledge Required (P1):

Building an effective medical store management system requires comprehensive knowledge of C programming and an understanding of both inventory systems and user experience design. Developers must implement key modules to manage product data, transactions, and secure record keeping.

- **Engineering Fundamentals (K3):**
  Structured programming principles are applied to support inventory management, transaction processing, and data searching. Developers use functions, structures, and control logic to model real-world pharmacy operations.
- **Engineering Specialist Knowledge (K4):**
  Domain-specific knowledge of pricing models, medicine sales flow, and stock updates is essential. Features such as tax calculation, secure payment integration, and purchase confirmation rely on pharmacy-specific operational logic.
- **Engineering Practice (K6):**
  Practical implementation includes file handling for storing medicine details and transaction logs, designing user-friendly input-output interfaces, and ensuring real-time updates during purchase operations. The system mimics standard retail software within a pharmacy context.

2. Depth of Analysis Required (P3):

Designing this system involves analysis of pharmacy workflows, medicine demand trends, and real-time transaction management. Developers must model medication pricing, inventory dependencies, and customer interaction logic to optimize usability and reliability. Careful planning is needed to handle edge cases such as out-of-stock scenarios or partial payments.

3. Extent of Stakeholder Involvement and Conflicting Requirements (P6):
Stakeholders include pharmacy owners, on-site staff, customers, and regulatory authorities. Pharmacy staff require efficient tools for updating inventory, customers expect a smooth and transparent purchasing experience, and regulatory compliance demands secure and accurate record management. Developers must communicate with all stakeholders to balance usability, operational needs, and legal compliance, resulting in a robust and user-aligned system.

Addressed Engineering Activities (A)

1. Range of Resources (A1):
The development and deployment of the Medical Store Management System involve a wide range of resources. People such as developers, pharmacy staff, and customers are key users of the system. Financial resources are needed for the development process, hardware setup, and ongoing support. Technical infrastructure includes computers and local servers used for data entry, transaction processing, and storage. Materials such as user manuals and documentation support system training and adoption. Informational assets like medicine details, pricing structures, inventory logs, and sales records form the foundation of the system. The technology stack is based on C programming with file handling techniques to enable secure storage and efficient execution. Combined, these resources contribute to a reliable and efficient pharmacy solution that improves service delivery and internal control.

Complex Engineering Problem 13

Vehicle Rental System

To enhance the efficiency and accuracy of vehicle rental operations, the Vehicle Rental System is developed using C programming as a digital platform to manage bookings, vehicle status, and billing. The system comprises two major modules: an administrative panel for internal use by rental agents and a customer interface for external interaction.

Through the admin panel, authorized staff can add, remove, or update vehicle records, including vehicle ID, category, rental rate, and availability status. Staff can also monitor ongoing rentals and process returns. The customer interface allows users to browse available vehicles, select rental periods, and view cost estimates. The system automatically calculates total rental charges based on duration, vehicle type, and any additional fees. Upon return and payment confirmation, vehicle availability is updated, and the rental history is logged. File handling in C ensures secure data storage for all booking and customer-related information.

This system minimizes human errors, eliminates overbooking issues, and enhances the overall rental experience for customers, while streamlining backend operations.

Addressed P and K

1. Depth of Knowledge Required (P1):
Developing this system requires strong command of C programming and understanding of rental operations and time-based billing models.

- **Engineering Fundamentals (K3):**
  The system utilizes functions, structures, and logical control flows to handle bookings, pricing, and availability tracking.
- **Engineering Specialist Knowledge (K4):**
  Domain-specific knowledge such as vehicle categories, late fee policies, and rental constraints are necessary for accurate system modeling.
- **Engineering Practice (K6):**
  Practical implementation includes secure file handling, modular user interfaces, and dynamic billing computations to simulate real-world rental workflows.

2. Depth of Analysis Required (P3):
Developers must analyze rental duration, overlapping bookings, and billing conditions to ensure accurate cost calculation and service consistency. Logical planning is required to handle reservation conflicts and late return scenarios.

3. Extent of Stakeholder Involvement and Conflicting Requirements (P6):
Stakeholders include rental agents, customers, and management. Customers prioritize availability

and transparency, while staff require control and reporting features. These varied requirements necessitate balanced system design and feedback loops.

Addressed Engineering Activities (A)

1. Range of Resources (A1):

The development and operation of the Vehicle Rental System involve a wide array of resources across human, financial, technological, and informational domains. Human resources include rental staff who interact with the system, developers responsible for building and maintaining the platform, and customers who access the booking interface. Financial resources are allocated for development costs, system deployment, infrastructure upgrades, and ongoing technical support. Equipment includes desktop terminals for office operations, mobile kiosks for vehicle returns, and receipt printers. Materials such as rental agreements, booking confirmations, and invoices are generated and managed. Informational assets like vehicle logs, customer details, rental duration, pricing policies, and late fee rules form the core database. Technology resources encompass C programming tools, version control systems, and file-based storage for secure and efficient data handling. All these resources work cohesively to ensure operational success, customer satisfaction, and business scalability.

Complex Engineering Problem 14

Clinic Appointment Scheduling System

To streamline patient-doctor interactions and minimize appointment conflicts, the Clinic Appointment Scheduling System is developed using C programming. It is designed to manage scheduling, patient registration, and doctor availability in a clinical setting.

The system includes an administrative module for clinic staff and an interface for patients. Through the admin panel, users can set doctor schedules, manage time slots, and register patient information. Patients can view available time slots and schedule appointments accordingly. The system prevents double bookings and automatically updates available slots. All appointment and patient records are securely stored using file handling in C programming.

This system enhances service delivery by automating scheduling, reducing wait times, and maintaining organized appointment logs.

Addressed P and K

1. Depth of Knowledge Required (P1):
Developers must understand data structuring, time-slot management, and clinical workflows to design an effective scheduling solution.

- **Engineering Fundamentals (K3):**
  Programming constructs like arrays, structures, and functions are used to handle time tracking and appointment management.
- **Engineering Specialist Knowledge (K4):**
  Understanding patient flow, doctor availability, and appointment rules is critical to accurate scheduling.
- **Engineering Practice (K6):**
  Includes designing user interfaces, handling dynamic input, and securely storing sensitive appointment data using file handling.

2. Depth of Analysis Required (P3):
Time conflict detection, concurrent bookings, and workflow optimization demand analytical thinking and logical system modeling.

3. Extent of Stakeholder Involvement and Conflicting Requirements (P6):
Stakeholders include doctors, patients, and reception staff. Their different expectations—efficiency, accessibility, and clarity—must be harmonized through the system design.

Addressed Engineering Activities (A)

1. Range of Resources (A1):

A diverse set of resources are essential for developing and operating the Clinic Appointment Scheduling System. These include clinic personnel such as doctors, receptionists, and nurses who depend on accurate scheduling to manage patient flow. Developers are involved in system design, implementation, and maintenance. Financial investment covers hardware procurement, software licensing, and technical support. Equipment includes computers, check-in terminals, and printers for appointment slips. Supporting materials involve appointment forms, patient intake documents, and scheduling sheets. Key information includes doctor availability, patient history, and appointment times. The system relies on C programming, secure file handling, and structured data formats to maintain integrity and support seamless operation of clinical schedules.

Complex Engineering Problem 15

Complaint Management System

To provide a systematic approach to logging, tracking, and resolving issues, the Complaint Management System is developed using C programming. The system enables users to file complaints, and administrators to manage, assign, and close them efficiently.

The system provides a complaint ID, category, description, and status for each issue logged. Admins can filter complaints, assign resolution teams, and update statuses. Users can check complaint status using their ID. All data is securely stored using file handling techniques in C.

This solution increases operational transparency, response efficiency, and long-term complaint tracking across institutions or organizations.

Addressed P and K

1. Depth of Knowledge Required (P1):
Developers need a sound understanding of C programming, record structuring, and user interface logic.

- **Engineering Fundamentals (K3):**
  Complaint records, status updates, and ID handling are implemented using structured data models and control statements.
- **Engineering Specialist Knowledge (K4):**
  Complaint resolution workflows, ticketing logic, and categorization require specialized domain understanding.
- **Engineering Practice (K6):**
  The use of file handling for storing and retrieving complaint logs reflects real-world support and ticketing platforms.

2. Depth of Analysis Required (P3):
Developers must analyze resolution workflows, prioritization schemes, and historical trends for optimal system logic.

3. Extent of Stakeholder Involvement and Conflicting Requirements (P6):
Stakeholders include users filing complaints, technicians resolving them, and admins tracking progress. Their expectations must be balanced through intuitive interfaces and efficient backend logic.

Addressed Engineering Activities (A)

1. Range of Resources (A1):
The Complaint Management System is supported by a wide range of resources. People include system users submitting complaints, developers maintaining the system, and administrative staff managing complaint resolution. Financial investments may include software tools, support services, and infrastructure costs. Equipment includes workstations for admin teams, logging terminals, and display dashboards. Materials involve complaint tracking sheets, reporting templates, and escalation guidelines. Core information includes complaint metadata, response history, user contact details, and resolution outcomes. Technologically, the system uses C programming for logic implementation and file systems to manage structured data, ensuring efficient issue tracking and service transparency.

Complex Engineering Problem 16

Hostel Management System

Managing student housing efficiently requires a structured solution. The Hostel Management System, developed in C programming, helps administrators manage room allocations, student records, and fee tracking.

The system maintains records of student names, room numbers, fee status, and complaint logs. Admins can allocate or reassign rooms, view available beds, manage payments, and generate reports. All operations and data, including complaints and service requests, are securely stored using file handling in C.

The system improves transparency, reduces manual errors, and facilitates smoother hostel operations.

Addressed P and K

1. Depth of Knowledge Required (P1):
The system requires skills in memory management, structured data, and allocation logic using C.

- **Engineering Fundamentals (K3):**
  Structured programming is used for student data entry, room tracking, and fee management.
- **Engineering Specialist Knowledge (K4):**
  Hostel-specific features like reallocation, visitor tracking, and complaint handling require domain knowledge.
- **Engineering Practice (K6):**
  Includes persistent data management through files and creating interfaces to manage hostel workflows effectively.

2. Depth of Analysis Required (P3):
Developers analyze vacancy rates, fee cycles, and complaint frequency to design logical modules for each function.

3. Extent of Stakeholder Involvement and Conflicting Requirements (P6):
Students, wardens, and accountants have varying needs. Efficient integration of these roles requires clear communication and inclusive system design.

Addressed Engineering Activities (A)

1. Range of Resources (A1):

The Hostel Management System draws upon a variety of resources to function effectively. These include human resources such as hostel wardens, support staff, and students. Financial inputs cover infrastructure, maintenance, and licensing costs. Equipment includes computers for check-in and billing, display boards for notices, and printers for receipts and records. Physical materials include registration forms, room assignment charts, and fee receipts. Informational resources consist of room availability records, student histories, payment logs, and maintenance requests. Technologically, the system is built with C programming and file handling capabilities to ensure consistent and secure operations.

Complex Engineering Problem 17

Parking Lot Management System

To organize vehicle access and slot utilization, the Parking Lot Management System is developed in C programming. It automates vehicle check-in/out, slot tracking, and billing.

Each vehicle is assigned a slot with entry time, type, and duration. Admins can check slot availability, view vehicle history, and calculate charges based on duration. Exit processing includes bill generation and slot release. All data is persistently stored using file handling.

This system enhances operational flow, maximizes slot usage, and reduces congestion.

Addressed P and K

1. Depth of Knowledge Required (P1):
Involves timing logic, slot availability checking, and billing algorithms in C.

- **Engineering Fundamentals (K3):**
  Time-based fee calculations and vehicle logs are managed using structures, arrays, and control flows.
- **Engineering Specialist Knowledge (K4):**
  Parking logistics, slot reuse policies, and billing cycles require specific domain insight.
- **Engineering Practice (K6):**
  Practical development includes real-time slot allocation, fee computation, and file-based vehicle history.

2. Depth of Analysis Required (P3):
Developers must analyze slot turnover, peak usage hours, and wait times to ensure smooth operation.

3. Extent of Stakeholder Involvement and Conflicting Requirements (P6):
Vehicle owners, parking staff, and supervisors must be considered. Conflicting needs for availability, speed, and oversight must be resolved through thoughtful design.

Addressed Engineering Activities (A)

1. Range of Resources (A1):

A wide spectrum of resources supports the Parking Lot Management System. Human resources include parking attendants, maintenance crews, developers, and end users. Financial investments cover infrastructure setup, security systems, and software development. Equipment includes gate control systems, computers, billing counters, and security cameras. Essential materials include printed tickets, slot maps, and signage. Informational data such as vehicle logs, timing records, and billing history are stored and analyzed. The system uses C programming and file-based logic to process real-time slot usage and generate performance analytics, contributing to an efficient and scalable parking solution.