1. Define the Object-Oriented Paradigm. Explain five key characteristics that distinguish it from the procedural paradigm.

OOP refers to an approach of programming that uses objects(instances of a class) to represent real world entities that contain attribute(data) and methods(behavior).

Characteristics that distinguish OOP and Procedural Paradigm

|  |  |  |
| --- | --- | --- |
|  | Object-oriented Paradigm (OOP) | Procedural Paradigm |
| 1. Encapsulation | Combines records (attributes) and capabilities (techniques) into items, proscribing direct access (statistics hiding). | Information and functions are awesome; no facts safety is enforced. |
| 2. Abstraction | Emphasizes key capabilities and conceals elaborate implementation information (e.g., interfaces/summary training). | Famous implementation info; features act immediately on facts. |
| 3. Inheritance | Allows hierarchical relationships between training (baby lessons inherit techniques/residences from parent classes). | No inheritance; code reuse through functions/libraries. |
| 4. Polymorphism | Allows methods to behave in another way relying at the kind of item (e.g., technique overriding/overloading). | Capabilities are constant; no dynamic behavior relying on context. |
| 5. Stateful objects | Gadgets encapsulate inner country (facts) and behavior (strategies) collectively. | Functions take records, which they manner with out retained country. |

1. Using a simple case study (e.g. Student Registration System), draw a Use Case diagram to represent the interaction between users and the system

Student

Admin

Professor

Generate Reports

Login

Create Course

Display Courses

Register Course

Drop Course

Modify Course

Grade the course

View Class Roster

1. From the use case in Question 2, identify and draw the corresponding Class Diagram. Include at least 4 classes with attributes and methods

|  |
| --- |
| Student |
| StudentID: String  Name: String  Email: String  Major: String |
| RegisterCourses()  DropCourses()  ViewGrades() |

|  |
| --- |
| Course |
| CourseId: String  Title: String  CF: Int |
| EnrollStudent()  UnenrollStudent() |

|  |
| --- |
| Department |
| DepartmentID: String  Name: String |
| CreateClassRoster() |

|  |
| --- |
| Professor |
| StaffId:String  Name: String |
| GradeCourses()  ViewClassRoster() |

1. Define and explain a State Model. Using the same system, create a State Diagram for a “Student” object showing various states (e.g. Registered, Enrolled, Suspended, Graduated).

Registered

Suspended

Graduated

Enrolled

New Applicant

submitApplication()

registerCourse()

CompleteCourse()

feePaymentFailed()