

UML Covers...

- * Structure Diagrams
 - * Class Diagram
 - * Object Diagram
 - * Package Diagram
 - * Composite Structure Diagram
 - * Component Diagram
 - * Deployment Diagram
 - * Profile Diagram

- * Behaviour Diagrams
 - * Use Case Diagram
 - * Information Flow Diagram
 - * Activity Diagram
 - * State Machine Diagram
 - * Interaction Diagram
 - * Sequence Diagram

- * And More!!!

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

We Will Use...

Assignment Project Exam Help

- * Use Case (Model) Diagram
- * Class Diagram
- * Sequence Diagram

<https://powcoder.com>

Add WeChat powcoder

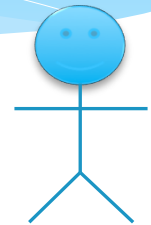
Use Case Diagram

Assignment Project Exam Help

- * Actor – Represents a role of a user of a system. The actor is not necessarily human.
- * Use Case – A usage scenario. One bubble per detailed scenario should be represented. Name can be in bubble or below.
- * Relationship – Shows a logical link between actor and use case. Use line or arrow.

<https://powcoder.com>

Add WeChat powcoder



Use Case Diagram - Advanced

Assignment Project Exam Help

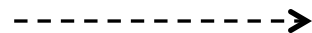
- * System Boundary – system domain limits between use case sets
- * Each box represents a part of the system and use cases falling within that system should be represented inside the box.
- * Inclusion – show relationships between use cases with common behaviour
- * Extension – show relationships with use cases that have the same goal but extra substeps

<https://powcoder.com>

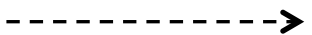
Add WeChat powcoder



<< include >>



<< extend >>



Class Diagram

Assignment Project Exam Help

- * Provide a description of the systems types and relationships between the components
<https://powcoder.com>
- * It is independent of the implementation
- * Allows software engineers to evaluate the design of the system before development

Add WeChat powcoder

UML Class Diagram

Assignment Project Exam Help

- * Describes each class in the system
 - * Class Name (bold)
 - * All attributes with Types
 - * All methods, including arguments and return types

ClassName
attributeName:type
methodName():returnType

- * Additional Details
 - * + public member
 - * # protected member
 - * - private member
 - * static member
 - * *abstract operation*

Student
+ studentID: Integer
- password: String
enroll(Course course): Boolean

<https://powcoder.com>

Add WeChat powcoder

Abstract Classes and Interfaces

Assignment Project Exam Help

- * Abstract Class names and methods should be *italic*
- * Interface classes should be labeled

Add WeChat powcoder

<i>ClassName</i>

<<interface>> InterfaceName

Extension and Implementation

Assignment Project Exam Help

- * Classes that extend another class

- * Show inheritance between classes

Add WeChat powcoder

- * Classes the implement another class

- * Show interfaces and classes



Multiplicity

Assignment Project Exam Help

* 0..1

Zero or One

* 1

<https://powcoder.com>

One Only

* 0..*

Zero or More

* 1..*

Add WeChat powcoder

One or More

* [Int]

Exactly a Number

* [Int]..[Int]

Exactly a Range (inclusive)

UML Sequence Diagram

Assignment Project Exam Help

- * Shows the order in which interactions should occur and the relationships between objects during execution
- * Shows communication between classes, which can aid in updating software and refactoring

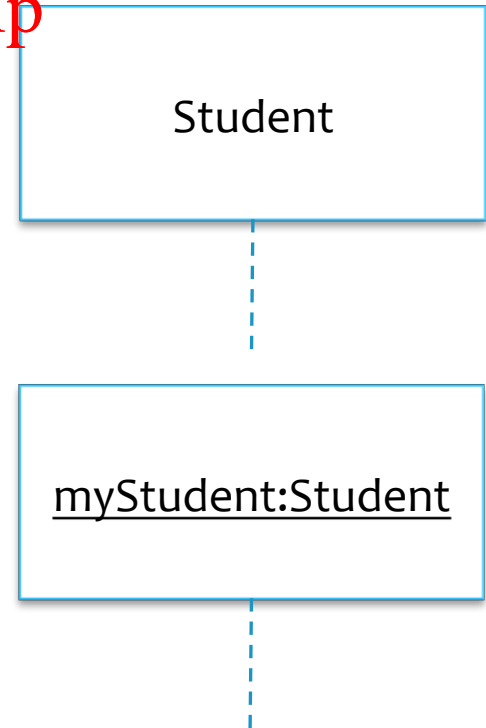
<https://powcoder.com>

Add WeChat powcoder

UML Sequence Diagrams



* Lifeline

- * This can represent a type of object or an instance of an object. Instances should be Underlined
- * This distinction is only needed if you have multiple named object of the same type



UML Sequence Diagrams

Assignment Project Exam Help

- * Messages – show function calls between objects
 - * Synchronous Messages 
 - * Asynchronous Messages 
- * The message represents a function or method in the receiving object
 - * The method signature should be noted above the message line
- * Return value is noted using a dotted line, with the value type/name written above the line

