UML 2 Intermediate Exam Overview

Exam Series Code	OMG-OCUP2-INT200
Exam Duration	105 minutes in native English-speaking countries and 135 minutes in all others. **Note**: When scheduling your exam in a non-native English-speaking country, you will not see this extra time until you complete your exam order.
Exam Fee	US$350 (or local equivalent)
Exam Type	Multiple choice (text and UML diagrams)
Exam Pass Score	>=51 of 90 questions answered correctly (>=56.6%)
Exam Prerequisite(s)	Passing Score on the OCUP 2 Foundation Certification exam.
Exam Specification	Unified Modeling Language (UML) v.2.5.1
Recommended Exam Study Guides	1. UML 2.0 in a Nutshell (Pitman)
	2. UML 2 for Dummies (Schardt)
Additional Reading	Model Organization with Packages and the Package Diagram (Baker)
	Concurrency in UML (Stachecki)
	Getting It Right on the Dot
Useful Knowledge	Modeling using UML, BPMN, SysML, or Realtime software development principles.
Exam Training Options (not required)	EduMAX
	NobleProg (**Worldwide**: Scroll to 'Other Countries' section to change region)
	oose (**Germany**: 2-day course and 3-day course)
Exam Voucher Program	View our Voucher Program for potential discounts.
Testing Accommodations	For hearing, learning, physical and/or visual disability accommodations, please contact certification@omg.org with proof of your most recent diagnosis thereof so we can officially request accommodation(s) on your behalf via Pearson, and once approved, provide you with further instructions on scheduling your exam(s) with your requested accommodation(s).
Exam Registration	Pearson VUE: create an account, locate a test center, view available tests, (re)schedule a test (online or at a test center), cancel your exam (contact Pearson VUE >24 hours prior to exam for a full refund or you forfeit the full exam price), view exam scores and Contact Pearson VUE (for any technical issues-**use chat feature to expedite a response**).
Online Exam Check-In & Requirements	Visit Pearson VUE Online Proctoring for detailed info. Log in at least 30 minutes early (online verification may take 15-20 minutes). Late arrivals will not be allowed to take the exam.
Test Center Check-In & Requirements	Arrive at least 30 minutes early. Late arrivals will not be allowed to take the exam. Bring two forms of ID (at least one with photo and both with signature): alien registration card, bank card, credit card, employee badge, government issued, green card, military, passport, school and state ID. Do not bring any items (personal or otherwise) other than the two forms of ID to a test center.
Technical Issues	Contact Pearson VUE (**use chat feature to expedite a response**).
Exam Languages	This exam is only offered in English. You cannot use a translation app during the exam.
Review Your Answers	Before completing your exam, you will be presented with a screen to review your answers to all questions.
Exam Score Reports	Whether at a test center or online, pass or fail, you will be provided with a score report on your computer screen immediately following your exam. A hardcopy of your score report will be provided before an individual leaves a test center with their score in each major section. You can also review your exam score reports via your Pearson VUE account. If you fail your exam, you can review general sections where you scored poorly to assist when you decide to retake your exam.

Digital Badges/Certificates	Those who pass their exam will immediately receive an email from Credly (admin@credly.com - check Junk folder) to claim their verifiable digital badge. Credly provides certified professionals with the option to share their certification credentials with others via the Credly Network, social media, print to .pdf or hardcopy certificate, and other avenues.
Retake Vouchers	If you failed your exam, contact certification@omg.org to request a 30% discounted exam retake voucher.
Certification Expiration	Your certification expires 5 years from the date you passed your exam. The same or a higher-level certification must be taken prior to the previous certification's expiration date to extend your certification.
Original UML Certification	While the original UML certification is still recognized by some, the UML 2 certification will demonstrate modeling knowledge and skills required in today's complex IT environment.
Still Have Questions?	certification@omg.org

General Areas Tested in the UML 2 Intermediate Exam

Activities & Actions	20%
Basic Structure	16%
Interactions	14%
State Machines & Use Cases	13%
Components	12%
Common Behavior	10%
Structured Classifiers	9%
Classification	6%
Total	**100%**

Comprehensive Areas Tested in the UML 2 Intermediate Exam

STRUCTURAL DIAGRAMS	**CLASSIFICATION**	**COMMON STRUCTURE**	**PACKAGES**	**SIMPLE CLASSIFIERS**	**STRUCTURED CLASSIFIERS**	**VALUES**	ADDITIONAL TOPICS	**DEPLOYMENT**
	Classifiers Generalization Sets Instances Operations Properties	Constraints Dependencies Namespaces Types & Multiplicity	Packages	Interfaces Signals	Associations Classes Collaborations Components Encapsulated Classifiers Receptions Structured Classifiers	Intervals Time		Artifacts Nodes
BEHAVIORAL DIAGRAMS	**ACTIONS**	**ACTIVITIES**	**COMMON BEHAVIOR**	**INTERACTIONS**	**STATE MACHINES**	**USE CASES**		**INFORMATION FLOWS**
	Actions Expansion Regions Invocation Actions Structured Actions	Activities Activity Groups Control Nodes Object Nodes	Events Behaviors	Communication Diagrams Fragments Interactions Interaction Uses Lifelines Messages Sequence Diagrams	ProtocolStateMachines StateMachine Topics	Use Cases		Information Flows

The following provides **UML 2 Intermediate** exam coverage. Please refer to the **Unified Modeling Language (UML) v.2.5.1** specification for a more in-depth look at the corresponding chapters and sections cited below.

CHAPTER 6: GENERAL TOPICS

Backus-Naur Form (BNF)

- This and the OCUP 2 Advanced exam use BNF where appropriate to specify textual notation, similar to the way it is used in the UML specification itself. BNF is defined in Ch. 6 of the UML 2.5 Specification (identically in the beta and formal versions). Also in Ch. 6 is a (very!) brief description of *execution scope*, a term that will be used later in several contexts.

Abstract Syntax

- Every first-level subsection of the UML specification starts with a UML diagram labeled *Abstract Syntax*. The OCUP 2 exams do not ask about these diagrams explicitly, but they are good examples of the language you're studying(!) and represent the relationships linking the elements to be presented in the sections that follow in a particularly clear and concise way. Learn to read them - this will provide an advantage to your study.

CHAPTER 7: COMMON STRUCTURE

- 7.3 Templates -
  - Templates are *Excluded* from Foundation and Intermediate levels; Templates and the many elements that support them will be covered at Advanced level. This exclusion encompasses elements and attributes defined for Templates here and later on (String Expressions and Name Expressions, e.g.; most have "Template" somewhere in their names). There are many of these scattered throughout the specification but we will not point out on the Foundation and Intermediate Coverage Maps, for each one, that it is excluded. This exclusion applies even within subsections denoted "*All*" in this coverage list.
- 7.4 Namespaces - *Add*:
  - ownedRule constraints, nested nameSpaces, circle-plus notation, ElementImport
- 7.5 Types and Multiplicity - *Add*:
  - Cardinality, isOrdered, isUnique, multiplicity string
- 7.6 Constraints - *Add*: Owner
  - In the exam, constraints will be expressed in simple OCL, possibly using Boolean expressions. Candidates should be able to read and understand these.
- 7.7 Dependencies - *Add*:
  - Usage, Abstraction

CHAPTER 8: VALUES

- 8.4 Time - *All*
- 8.5 Intervals - *All*

CHAPTER 9: CLASSIFICATION

- 9.2 Classifiers
  - 9.2.3 Semantics
    - Classifiers: *Add* Redefinition *Except* redefinitionContext

- 9.2.4 Notation: NOTE: UML allows a conforming tool to suppress the drawing of individual compartments or features of a classifier. Scenarios in the Intermediate and Advanced examinations may use this ability.
- 9.5 Properties
  - 9.5.3 Semantics
    - *Add* Properties as memberEnds of Associations, and the semantics of the defaultValue, properties isStatic and isDerived.
- 9.6 Operations
  - 9.6.3 Semantics
    - *Add* Constraints (preconditions, postconditions, bodyCondition)
- 9.7 Generalization Sets - *All Except* powertypes
- 9.8 Instances
  - 9.8.3 Semantics
    - *Add* InstanceSpecification partially representing the instance it corresponds to, classification of the instance by zero or more than one Classifier, type restrictions on a defining ValueSpecification, and snapshots

CHAPTER 10: SIMPLE CLASSIFIERS

- 10.3.3 Semantics
  - Signals - *All*
  - Receptions - *All*
- 10.4 Interfaces - *Add* ownership of a ProtocolStateMachine

CHAPTER 11: STRUCTURED CLASSIFIERS

- 11.1 Summary - *All*
- 11.2 Structured Classifiers
  - 11.2.1 Summary. Note that, because StructuredClassifier is abstract, covered aspects will be tested in the context of derived concrete metaclasses.
  - 11.2.3 Semantics
    - ConnectableElement: *All*
    - Parts and Roles - *All*
    - Connectors - *All Except* contracts
    - Multiplicities and topologies - *All except* n-ary Connectors
- 11.3 Encapsulated Classifiers
  - 11.3.3 Semantics
    - Ports - *All*
- 11.4 Classes
  - 11.4.3 Semantics
    - Classes: *Add* detailed aspects of attributes, namespaces, isActive
- 11.5 Associations
  - 11.5.1 Summary: *Add* AssociationClass
  - 11.5.3 Semantics
    - Associations: *Add* navigableOwnedEnd. NOTE that the dot notation signifying ownership of an association end by an associated Classifier, new in UML 2.5, will be covered. See the Additional Reading section in the first table above on this sheet.
    - AssociationClass: *All Except* Class: ownedAttribute and Association: ownedEnd
- 11.6 Components
  - 11.6.1 Summary: *All Except* modeling Components through the development life cycle (which is methodology-dependent and so not covered in OCUP 2) and profiles (covered in Advanced)

    - 11.6.3 Semantics " Components: *All Except* details about wiring dependency, details of the "white-box" view (although candidates should be *aware* of the white-box view), execution time semantics of a Connector, and «Specification» and «Realization» stereotypes
- 11.7 Collaborations
    - *All Except* specializing collaborations, roleBindings, Connector details, representation

CHAPTER 12: PACKAGES

- 12.2 Packages
    - 12.2.3 Semantics
        - Package: *Add* specifying the URI
        - Model: *All*

CHAPTER 13: COMMON BEHAVIOR

- 13.1 Summary: *All*
- 13.2 Behaviors
    - 13.2.3 Semantics:
        - Behaviors: *Add* Behavior as a class
        - Behavior Parameters: *Add* defaultValue, streaming (complete at this level)
        - Opaque and Function Behaviors: *Includes* OpaqueBehavior (only)
        - Behaviored Classifiers: *All except* the distinction between ownership as a nested classifier compared to ownedBehavior, and precise semantics of classifierBehavior
        - Behavioral Features and Methods: *Add* method, context, parameters
- 13.3 Events
    - 13.3.1 Summary: *All*
    - 13.3.3 Semantics
        - Event Dispatching: Includes Event and Trigger. Excludes SignalBroadcastAction, event pool
        - Message Events: *All except* SignalBroadcastAction.
        - Change Events: *All*
        - Time Events: *All*

CHAPTER 14: STATEMACHINES

- StateMachine coverage at Intermediate level:
    - *Add* specification of a method of a behavioredClassifier (that is, an Operation or Reception corresponding to a BehavioralFeature); regions; vertices; submachine State; history (deep or shallow); deferred events; the pseudostates join, fork, entrypoint, exitpoint, and terminate; transition kind=local; high-level (group) transitions; conflicting transitions; firing priorities; transition selection and execution sequence.
    - Also *Add* ProtocolStateMachine, *Except* Declarative and Executable interpretation, multiple ProtocolStateMachines per Classifier, and Protocol Conformance.
    - NOTE: Unexpected trigger reception and Unexpected behavior will not be covered in OCUP 2.

CHAPTER 15: ACTIVITIES

- 15.2 Activities
    - 15.2.3 Semantics:

        - Activities: *Add* the null token, token movement details resulting from offer and acceptance, named edges, Activities as classes.
        - Activity Nodes: *Add* concurrent execution, one token offered to multiple targets
        - Activity Edges: *Add* token ordering
        - Object Flows: *Add* null token.
        - Activity Execution: *Add* Parameters, behavior at first invocation.
    - 15.2.4 Notation: *Add* eliding pins, connectors, class notation
- 15.3 Control Nodes
    - 15.3.3 Semantics
        - Initial Node: *Add* additional concurrent flows and CentralBufferNodes
        - Final Nodes: *Add* isSingleExecution
        - Fork Nodes: *Add* handling of unaccepted token offers
        - Join Nodes: *Add* joinSpec and isCombinedDuplicate
        - Decision Nodes: *Add* decisionInputFlow and the primary incoming edge. Exclude decisionInput behavior and guards on multiple outgoing edges.
    - 15.3.4 Notation
        - Combined MergeNode and DecisionNode
- 15.4 Object Nodes
    - 15.4.1 Summary: *Add* CentralBufferNodes and DataStoreNodes
    - 15.4.3 Semantics:
        - Object Nodes: *Add* Multiple object tokens with the same value, ObjectNode's type, instate
        - CentralBufferNodes: *All*
        - DataStoreNodes: *All Except* selection, transformation
- 15.6 Activity Groups
    - 15.6.3 Semantics
        - Activity Partitions: *All Except* the descriptive text about preparation of descriptive models for review
        - Interruptible Activity Regions: *All Except* isSingleExecution

CHAPTER 16: ACTIONS

- 16.1 Summary: *Add* Actions as Interactions, and as contained in Behaviors.
- 16.2 Actions:
    - 16.2.3 Semantics
        - Actions: *Add* context BehavioredClassifier, StructuredActivityNodes, streaming, multiple instances, and effects of violations of localPrecondition and localPostcondition.
        - Opaque Actions: *Add* interpretation of body strings
        - Pins: *Add* multiplicity requirements on output pins for termination, ValuePins and ActionInputPins do not enable Action execution count, ValuePin, ActionInputPin, but *Exclude* fromActions.
        - Actions and Pins in Activities: *Includes* basic semantics of Actions and Activities, input token requirements for execution and output requirements on completion, *Except for* disallowing of acceptance of more tokens than will be consumed by one execution of an Action, isLocallyReentrant, isControl, isControlType.
- 16.3 Invocation Actions
    - 16.3.3 Semantics
        - Call Actions: *Add* synchronous or asynchronous invocation behavior, passing and returning values, streaming
- 16.5 - 16.9: Material in these sections is not covered in OCUP 2.
- 16.10 Accept Event Actions
    - 16.10.3 Semantics

        - UnmarshallAction will not be tested in OCUP 2
- 16.11 Structured Actions
    - 16.11.1 Summary: Basic definitions, *Excluding* ConditionalNodes, LoopNodes, and SequenceNodes which will not be tested in OCUP 2
    - 16.11.3 Semantics
        - Structured Activity Nodes: *All Except* Variables, semantics of activity edge when contained or not contained by a StructuredActivityNode.
- 16.12 Expansion Regions
    - 16.12.1 Summary: *All* (the basic definition)
    - 16.12.3 Semantics: *All Except* Execution Engine defining collection types

CHAPTER 17: INTERACTIONS

- 17.1 Summary
    - 17.1.2 Basic Trace Model: *Add* Interaction equivalence. Disallowed or invalid traces will be tested at Advanced level only. Some incidental references to invalid traces may be included in sections specified here; nevertheless, this concept will be tested at Advanced level only.
    - 17.1.3 Partial ordering constraints on valid traces: *Add* coregion or parallel operator effect
    - 17.1.4 Interaction Diagram Variants: The *Sequence Diagram* was tested at Foundation level. The *Communication Diagram* is tested at Intermediate level. The *Interaction Overview Diagram* will be tested at Advanced level. Neither the *Timing Diagram* nor Interaction Tables will be tested in OCUP 2.
- 17.2 Interactions
    - 17.2.3 Semantics
        - Interactions: *Add* the use of a formal Gate
        - Interaction Fragments: *All*
        - State Invariants: *All*
- 17.3 Lifelines
    - 17.3.3 Semantics
        - Lifelines: *Add* parallel combined fragment
- 17.4 Messages
    - 17.4.3 Semantics
        - Messages: *Add* semantics as defined, lost and found messages, message arguments *Except* wildcard
        - Gates: *Add* all content not already covered at Foundation level
- 17.6 Fragments
    - 17.6.3 Semantics
        - *Include* Interaction Operands, Interaction Constraints, Combined Fragments, Interaction Operator Kind Values, Alternatives, Option, Break, Strict Sequencing, and Loop
- 17.7 Interaction Uses
    - 17.7.3 Semantics
        - *Include* Interaction Uses, Part Decompositions. NOTE: In Notation, <collaboration-use>, strict, and return value will be tested at Advanced level.
- 17.8 Sequence Diagrams
    - 17.8.1: Sequence Diagram Notation
        - Graphic Nodes: *Add* InteractionUse, CombinedFragment, StateInvariant, DurationConstraint DurationObservation, TimeConstraint TimeObservation
        - Graphic Paths: *Add* LostMessage, FoundMessage
- 17.9 Communication Diagrams
    - Introduction: *All*
    - 17.9.1: Communication Diagram Notation

        - NOTE that Frame, Lifeline, and Message were introduced at Foundation, and refer to the same definitions as for Sequence Diagrams except that Message refers also to 17.9.1 Sequence Expression, next:
        - Sequence Expression: *All Except* concurrent execution

CHAPTER 18: USE CASES

- 18.1 UseCases
    - 18.1.3 Semantics

Use Cases and Actors: *Add* description through a Collaboration; being owned by a Classifier.

CHAPTER 19: DEPLOYMENTS

- 19.1 Summary: *All Except* extending the package
- 19.2 Deployments
    - 19.2.1 Summary: *All*
    - 19.2.3 Semantics: *All Except* extending in profiles, Property and InstanceSpecification as targets
- 19.3 Artifacts
    - 19.3.1 Summary: *All*
    - 19.3.3 Semantics: Basic definition. Excludes organizing into composition hierarchies, extending especially as profiles (which will be tested at Advanced level)
- 19.4 Nodes
    - 19.4.1 Summary: *All*
    - 19.4.3 Semantics: *All*

CHAPTER 20: INFORMATION FLOWS

- 20.1 InformationFlows: Basic definition, uses, and notation