<table>
<thead>
<tr>
<th><strong>Exam Series Code</strong></th>
<th>OMG-OCUP2-ADV300</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Exam Duration</strong></td>
<td>105 minutes in English-speaking countries and 135 minutes in all others.</td>
</tr>
<tr>
<td><strong>Exam Fee</strong></td>
<td>US$250 (or local equivalent) in English-speaking countries and US$260 (or local equivalent) in all others.</td>
</tr>
<tr>
<td><strong>Exam Type</strong></td>
<td>Multiple choice (text and UML diagrams)</td>
</tr>
<tr>
<td><strong>Exam Pass Score</strong></td>
<td>&gt;=57 of 90 questions answered correctly (&gt;=63%)</td>
</tr>
<tr>
<td><strong>Exam Prerequisite(s)</strong></td>
<td>OCUP 2 Foundation and OCUP 2 Intermediate Certifications</td>
</tr>
<tr>
<td><strong>Exam Specification</strong></td>
<td>Unified Modeling Language (UML) v. 2.5.1</td>
</tr>
</tbody>
</table>
| **Recommended Exam Study Guides** | 1. UML 2.0 in a Nutshell (Pitman)  
2. UML 2 for Dummies (Schardt) |
| **Exam Languages**   | This exam is offered in English. Individuals cannot use a translation app during the exam. |
| **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). |
| **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. Pearson VUE Test Center Coronavirus Guidelines |
| **Exam Score Reports** | Pass or fail, individuals will be provided with a score report on computer screen immediately following the exam whether on-site at test center or online. A hardcopy will be provided before an individual leaves a test center with their score in each major section. If an individual fails, they can review those sections where they scored poorly to assist them when they decide to retake the exam. Individuals can also review their exam score reports via their Pearson VUE account. |

**Useful Knowledge**

- Modeling using UML, BPMN, SysML, or Realtime software development principles.

**Exam Training Options (not required)**

- NobleProg (Canada, China, Germany, India, North America, Poland, UAE and UK)

**Exam Voucher Program**

Visit the Pearson VUE Voucher Store for a 10% discount/10 vouchers or view our Voucher Program for greater discounts. Vouchers expire one year after purchase and can be transferred. Contact certification@omg.org for instructions on testing accommodations before registering for an exam.
Certification Digital Badges

Those who pass their exam will immediately receive an email from Credly (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, .pdf or hardcopy certificate, and other avenues.

Certification Expiration

Certifications expire 5 years from the date the exam was passed. The same or a higher-level certification must be taken prior to the previous certification's expiration date to extend a certification.

Retaking the Exam

Contact certification@omg.org to request a 30% discounted exam retake voucher.

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 Advanced Exam

<table>
<thead>
<tr>
<th>Area</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common Structure</td>
<td>21%</td>
</tr>
<tr>
<td>Classification</td>
<td>14%</td>
</tr>
<tr>
<td>The MOF &amp; Metamodeling</td>
<td>12%</td>
</tr>
<tr>
<td>Activities</td>
<td>9%</td>
</tr>
<tr>
<td>Interactions</td>
<td>9%</td>
</tr>
<tr>
<td>Structured Classifiers</td>
<td>8%</td>
</tr>
<tr>
<td>Actions</td>
<td>7%</td>
</tr>
<tr>
<td>Alf</td>
<td>6%</td>
</tr>
<tr>
<td>fUML</td>
<td>6%</td>
</tr>
<tr>
<td>StateMachines</td>
<td>6%</td>
</tr>
<tr>
<td>Common Behavior</td>
<td>3%</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>100%</strong></td>
</tr>
</tbody>
</table>

Comprehensive Areas Tested in the UML 2 Advanced Exam

<table>
<thead>
<tr>
<th>STRUCTURAL DIAGRAMS</th>
<th>CLASSIFICATION</th>
<th>COMMON STRUCTURE</th>
<th>STRUCTURED CLASSIFIERS</th>
<th>VALUES</th>
<th>DEPLOYMENTS</th>
<th>METAMODELING</th>
</tr>
</thead>
</table>
| Classifiers         | Classifier Templates | Information Flows | Associations | String Expression | Artifacts | Alf Integration with UML Model 
|                     | Features         | Name Expression | Collaborations | Values         | Deployments | Lexical Structure |
|                     | Generalization Sets | Profiles        | Components          | |
|                     | Operations       | Realization     | |
|                     | Properties       | Templates       | |
| ACTIONS             | AcceptEventActions | Activities      | Behaviors - Reentrant | Behavior StateMachines | protocol StateMachines |
|                     | Invocation Actions | Activity Groups | Events - Event Pool | StateMachine Redefinition |
|                     | Structured Actions - RaisedExceptionActions | Control Nodes | FunctionBehavior | |
|                     | Object Actions - ValueSpecificationActions | Exception | |
|                     | Object Nodes     | Executable Nodes | |
| BEHAVIORAL DIAGRAMS |                     |                  | |
|                     |                  |                  | |
|                     |                  |                  | |

The MOF and Metamodeling

Architectural alignment

Metamodels

Models

Models and what they model

Semantics of languages

The MOF
The following provides **UML 2 Advanced** 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 exam uses 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 on page 9 (UML 2.5 Specification, Beta 1). 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. As an Advanced candidate, you presumably know how to read these diagrams and use the information they display. If you don't have this skill, you should develop it. It will provide an advantage to your study, and your work in the field at this level.

**CHAPTER 7: COMMON STRUCTURE**

- 7.3 Templates - 
  - Add Templates. Postponed until now, Templates and the many elements that support them are covered at this Advanced level. Coverage is fairly complete, encompassing elements and attributes defined for Templates here in Section 7.3 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, for each, that it is now included. We will, however, specifically mention the following:
    - Add Template Signatures, Template Bindings, Bound Element Semantics, and Template Notation

- 7.4 Namespaces - Add:
  - NamedElement association with StringExpression, and having both a name and a nameExpression.

- 7.7 Dependencies - Add:
  - Realization

**CHAPTER 8: VALUES**

- 8.3 Add: String Expressions

**CHAPTER 9: CLASSIFICATION**

- 9.2 Classifiers
  - Add Classifiers: Add Classifier may own CollaborationUses and UseCases
  - Generalization: Add Substitutability
  - Redefinition: Add redefinitionContext
  - Substitution: Add

- 9.2.4 Notation: NOTE: UML allows a conforming tool to suppress the drawing of individual compartments or features of a classifier. Scenarios in this examination may use this ability.

- 9.3 Classifier Templates: Add
• 9.4 Features: Add: concurrency property, effect property, notation of feature redefinitions
• 9.5 Properties: Add: Note the reference to qualifiers. Add ternary and higher-order associations, redefinition, composition and transitive deletion, subsetted property, isDerivedUnion.
• 9.6 Operations: Add featuringClassifier, isQuery, owningClassifier context
• 9.7 Generalization Sets - Add: powertypes

CHAPTER 11: STRUCTURED CLASSIFIERS

• 11.2 Structured Classifiers: Add: contracts, n-ary Connectors
• 11.4 Classes: Add: the stereotype «Metaclass»
• 11.5 Associations: Add: n-ary Associations (n>2), Subsetting, Specialization, qualifiers and qualified Association end, derivation of an Association, navigability via Class:ownedAttribute and Association:ownedEnd
• 11.6 Components: Add: Profiles based around components, wiring dependency, details of the "white-box" view beyond the treatment at Intermediate level, execution time semantics of a Connector, and «Specification» and «Realization» stereotypes
• 11.7 Collaborations: Add: extension of collaborationRole in a specialization

CHAPTER 12: PACKAGES

• 12.3 Profiles: Includes All except MOF-equivalent semantics and non-UML metamodels. Also exclude XMI Serialization.

CHAPTER 13: COMMON BEHAVIOR

• 13.2 Behaviors Add: reentrant Behavior, Function Behavior, Behavior owned as a nestedClassifier
• 13.3 Events Add: Event handling by context object, event pool, wait point, SignalBroadcastAction

CHAPTER 14: STATEMACHINES

• 14.2 Behavior StateMachines: Add: event pool
• 14.3 StateMachine Redefinition: All

CHAPTER 15: ACTIVITIES

• 15.2 Activities:
  o Activities and Activity Nodes: Add: isControlType
  o Activity Edges: Add: Object tokens flowing over ControlFlow edges, object tokens accepted by ExecutableNodes, managing contention between multiple nodes, the weight property
  o Object Flows: Add: remainder of subsection. (Basic definition and null token already covered.)
  o Variables: All Except the discussion of variable setting in the Note paragraph.
  o Activity Execution: Add: remainder of subsection. (Material preceding isSingleExecution has already been covered.)
  o Activity Generalization: All.
• 15.3 Control Nodes
Decision Nodes: Add: decision input behavior, Parameters, and guards on multiple outgoing edges.

15.4 Object Nodes
- Object Nodes: Add upperBound, ordering, selection behavior
- Activity Parameter Nodes: Add: effect of ordering
- Data Store Nodes: Add: selection and transformation

15.5 Executable Nodes
- Executable Nodes: Add: concurrent execution
- Exceptions and Exception Handlers: All

15.6 Activity Groups
- Activity Partitions: Add: the descriptive text about preparation of descriptive models for review
- Interruptible Activity Regions: Add: isSingleExecution

CHAPTER 16: ACTIONS

16.1 Summary: Add dependence of Actions on Activities, basic definition of concrete syntax, and of execution engine
16.2 Actions:
- Actions: Add isLocallyReentrant and isReentrant.
- Pins: Add ordering and isOrdered, token behavior on StructuredActivityNodes, fromActions
- Actions and Pins in Activities: Add: disallowing of acceptance of more tokens than will be consumed by one execution of an Action, isLocallyReentrant, isControl, isControlType.

16.3 Invocation Actions
- Call Actions: Add StartObjectBehaviorAction, classifierBehavior, non-reentrant and reentrant Behavior, matching owned Parametersto Pins by ordering
- Send Actions: BroadcastSignalAction, SendObjectAction, ordering of owned and inherited Properties of a Signal, effects of local or remote target object.
- Invocation Actions and Ports: All

16.4 Object Actions
- Summary: All
- ValueSpecificationAction: All

16.5 - 16.9: Material in these sections is not covered in OCUP 2.

16.10 Accept Event Actions
- Accept Call Actions: Add triggering by an asynchronous call, method behavior caveat
- Reply Actions: All

16.11 Structured Actions
- Structured Activity Nodes: Add: Variables, semantics of activity edge when contained or not contained by a StructuredActivityNode
- Isolation: All

16.13 Other Actions
- Raise Exception Actions: All

CHAPTER 17: INTERACTIONS

17.1 Summary
- Interactions in detailed design phase, all discussion of role of interactions, interleaving
- NOTE THAT ALL discussion of disallowed or invalid traces in this chapter is included. This Coverage Map does not list specific references to disallowed or invalid traces.
- Interaction Diagram Variants: Add Interaction Overview Diagram

17.2 Interactions
- Add Specializing and redefining an Interaction
CHAPTER 17: LIFELINES

17.3 Lifelines
- Add coregion

17.4 Messages
- Add representation of ConnectableElement with a Type, wildcard argument
- Messages: Add assignment-target, value-specification
- Notation: As in all other sections, notation of covered elements is included automatically. For this subsection, which includes some notation for elements not mentioned previously, we point out that All of the notation section is included.

17.5 Occurrences
- General Orderings: All

17.6 Fragments
- Consider Ignore Fragments: All
- Continuations: All
- Negative: All
- Critical Region: All
- Ignore/Consider: All
- Assertion: All

17.7 Interaction Uses
- Notation: InteractionUse, CollaborationUse, strict, and return value

17.8 Sequence Diagrams
- Sequence Diagram Notation: Add Continuation, coregion
- Graphic Paths: Add GeneralOrdering

17.9 Communication Diagrams
- Sequence expression: Add iteration notation for concurrent execution

17.10 Interaction Overview Diagrams: All

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: Add: extending the package
19.2 Deployments Add: extending in profiles, Property and InstanceSpecification as targets
19.3 Artifacts Add: organizing into composition hierarchies, extending especially as profiles

CHAPTER 20: INFORMATION FLOWS

20.1 InformationFlows
- Add InformationFlow sources and targets, channels, InformationItems

COVERAGE OF METAMODELING INCLUDES THESE TOPICS:

Our coverage of metamodeling and the functionality that it enables (executable UML, e.g.) is intended as a survey, and the experts who wrote the exam questions did not expect you to study these specifications in enough depth to be able to work with the language. Learn the basics of these topics well and try to retain this knowledge as your
modeling work evolves so that, when you come to a point in a project that calls for metamodeling or generating a UML model intended for execution, you know where to look for solutions.

Metamodeling and the MOF, from **UML 2.5.1**:

- 6.2 Architectural alignment: *All*
- 6.3.1 Models and What They Model: *All except Execution Scope*, which was covered in the main exam

Metamodeling and the MOF, from **fUML v1.5**:

- 6.2 On the Semantics of Languages and Models: *All*
- 6.3 On the Semantics of Metamodels: *All*

Metamodeling and the MOF, from **OMG White Paper**:

- Meta-Modeling and the OMG Meta Object Facility (MOF): *All*

Metamodeling and the MOF, from the **MOF 2.5 specification**:

- 9.1, 9.2 Reflection: *All*
- 10.1, 10.2 Identifiers: *All*
- 11.1, 11.2 Extension: *All*

Semantics of a Foundational Subset for Executable UML Models (fUML), from **fUML v1.5**:

- 1 Scope: *All*
- 4 Terms and Definitions: *All*
- 7.1 Abstract Syntax: Overview
- 8.1 Execution Model: Overview, Behavioral Semantics

**Action Language for Foundational UML (Alf), from Alf v1.1**:

- 1 Scope: *All*
- 2.3 Semantic Conformance: *All*
- 6.1 Overview - General: *All*
- 6.2 Integration with UML Models: *All*
- 6.4 Lexical Structure: *All*