

## SUBJECT DESCRIPTION FORM

---

**Subject Title:** Object-oriented Methods for Information System Development

**Subject Code:** COMP 316

**Number of Credits:** 3

**Hours Assigned:**   Lecture                   35 hours  
                                  Tutorial/Lab/Sem   14 hours

---

**Pre-requisite:** COMP 201, COMP 302 (Nil for 61025)

**Co-requisite:** Nil

**Exclusion:** COMP 314, COMP 414

---

### Objectives:

- To introduce students to the concepts and practices of the object-oriented approach to software development.
- To familiarize students with the tools and languages for object-oriented analysis, design and implementation.
- To expose students to the applications of object-oriented technologies.

### Student Learning Outcomes:

After taking this subject, the students should be able to:

#### Professional/academic knowledge and skills

- (1) analyze a problem using an object-oriented approach;
- (2) design an object-oriented model for a problem, and implement the design using appropriate object-oriented tools and techniques;
- (3) document the analysis and design of an information system using UML;

#### Attributes for all-roundedness

- (4) analyze and solve information system problems in a systematic manner;
  - (5) cooperate with team members in problem solving, report and present the solution to an information system problem clearly.
- 

### Alignment of Programme Outcomes:

Programme Outcome 1: communicate effectively in Chinese and English at a level sufficient for project and system presentation and documentation. This subject teaches elements of this outcome and provides practice for the students on this outcome as well as providing an opportunity to measure parts of the outcome.

Programme Outcome 2: demonstrate a global outlook in factors that can affect the way computing systems are developed and used. This subject teaches elements of this outcome and provides practice for the students on this outcome as well as providing an opportunity to measure parts of the outcome.

Programme Outcome 4: think and reason critically on developing alternatives in problem solving and application development, and be able to design and evaluate for the proper

solution by applying computing and related technologies. This subject teaches elements of this outcome and provides practice for the students on this outcome.

Programme Outcome 5: possess technical knowledge needed to solve computing problems and to realize solutions in programming and associated technology. This subject teaches elements of this outcome and provides practice for the students on this outcome as well as providing an opportunity to measure parts of the outcome.

Programme Outcome 6: be responsive to and follow closely the advancement in information technology and their impact to the industrial need for information technology, with an attitude of continuous and lifelong learning. This subject teaches elements of this outcome and provides practice for the students on this outcome.

Programme Outcome 7: work together as a team in project design and development, while exhibiting leadership in a group or team whenever designated or necessary. This subject provides practice for the students on this outcome as well as providing an opportunity to measure parts of the outcome.

### Syllabus:

Topic	Duration of Lectures
<b>1. Information system analysis and design</b> System Development Life Cycle: requirement definition and specification; analysis and design; implementation; testing.	5
<b>2. Object-oriented paradigm</b> Rationale for the OO approach; OO software development life cycle; objects and classes; abstraction, aggregation, messages, services, encapsulation, association, generalization, inheritance hierarchy, polymorphism, metaclass.	5
<b>3. Object-oriented analysis and design</b> Object behaviour analysis; identification of classes and objects in a problem domain; object relationships; identification of structures, attributes instance connection, services and message connections; design issues in problem domain: data management, task management and human interface components, mapping an OO data model into a relational model; reuse, patterns, and frameworks; Unified Modeling Language (UML).	15
<b>4. Object-oriented languages and systems</b> Characteristics of object-oriented tools such as C++, Java, Smalltalk.	5
<b>5. Object-oriented Project Management</b> Applications in information system analysis, design and software engineering; user interfaces, system conversion, user training, user guide and operations manual, system development planning and scheduling, software cost estimation.	5

<b>Total</b>	<b>35</b>
--------------	-----------

**Tutorials/Laboratories/Project presentations and seminars:**

Topic	Duration of Laboratory
1. OO/UML development tools such as Rational Rose. 2. Class exercises on OO analysis and design problems. 3. Presentations and demonstrations of projects.	14
<b>Total</b>	<b>14</b>

**Case Study: Nil**

**Method of Assessment:**

Continuous Assessment	60%
Individual exercises, tests, group project involving implementation, demonstration and presentation	
Examination	40%

**Method of Assessment for Learning Outcomes:**

Assessment method / task	% weighting	Intended subject learning outcomes to be assessed (Please check as appropriate)									
		1	2	3	4	5					
Assignments	60	x	x	x							
Lab exercises											
Project		x	x	x	x	x					
Mid-term		x	x	x							
Examination	40	x	x								
<b>Total</b>	<b>100</b>										

**Reference Books:**

1. Timothy C. Lethbridge and Robert Laganieri. Object-Oriented Software Engineering – Practical software development using UML and Java, McGraw-Hill, 2nd Edition, 2005.
2. Simon Bennett, Steve McRobb and Ray Farmer. Object-Oriented System Analysis and Design Using UML, 2nd Edition, McGraw-Hill, 2002.
3. Wolfgang Emmerich. Engineering Distributed Objects, John Wiley & Sons, 2000.

4. Grady Booch, James Rumbaugh and Ivar Jacobson. The Unified Modeling Language User Guide, Addison Wesley, 1999.
5. Grady Booch. Object-Oriented Analysis and Design with Applications, 3rd Edition, Benjamin/Cummings, 2005.
6. B. Meyer. Object-Oriented Software Construction, 2nd Edition, Prentice Hall, 2002.
7. M. Fowler and K. Scott. UML Distilled: A Brief Guide to the Standard Object Modeling Language, 3rd Edition, Addison Wesley, 2003.
8. B. Hughes and M. Cotterell. Software Project Management, 3rd Edition, McGraw-Hill, 2002.
9. Deitel & Deitel. Java: How to Program, 6th Ed., Prentice Hall, 2004.
10. Deitel & Deitel. C++: How to Program, 5th Ed., Prentice Hall, 2005.