

## Subject Description Form

<b>Subject Code</b>	COMP 232		
<b>Subject Title</b>	Fundamentals of Programming		
<b>Credit Value</b>	3		
<b>Level</b>	2		
<b>Pre-requisite / Co-requisite/ Exclusion</b>	Nil		
<b>Objectives</b>	<p>This subject allows students to:</p> <ul style="list-style-type: none"> <li>• obtain knowledge on the fundamental elements in computer programming.</li> <li>• learn advanced computer programming techniques necessary for developing more sophisticated computer application programs.</li> </ul>		
<b>Intended Learning Outcomes</b>	<p>Upon completion of the subject, students will be able to:</p> <p><i>Professional/academic knowledge and skills</i></p> <p>(1) understand the programming elements for solving computing related problems;</p> <p>(2) possess the ability to use various software tools to design and develop efficient computer programs for solving problems in a stepwise manner;</p> <p>(3) possess the ability to learn independently other high level programming languages.</p> <p><i>Attributes for all-roundedness</i></p> <p>(4) develop skills in problem solving using systematic approaches;</p> <p>(5) identify and develop problem solutions in a logical manner.</p>		
<b>Subject Synopsis/ Indicative Syllabus</b>	<b>Topic</b>	<b>Duration of Lectures</b>	
		<b>Lectures</b>	<b>Lab/Tut</b>
	<b>1. Introduction</b> Basic concepts.	2	1
	<b>2. Fundamentals of computer programming</b> Primitive data types and operations.	2	1
	<b>3. Flow controls</b> Selection constructs, repetition constructs, methods (functions).	6	3
	<b>4. Object-oriented programming</b> Classes and instances, constructors and finalizers, static methods, static variables and others.	5	3
<b>5. Strings</b>	2	1	

Character strings, string methods.		
<b>6. Arrays</b> List of items, ordered and indexed lists.	2	1
<b>7. Advanced object-oriented programming techniques</b> Inheritance, polymorphism, abstract classes, interfaces.	5	2
<b>8. Event-driven programming and GUI</b> Events, listeners, graphical user interface.	2	1
<b>9. Exception handling</b> Exceptions and errors, handlers.	2	1
<b>Total</b>	<b>28</b>	<b>14</b>

**Laboratory Experiment:**

This subject emphasizes both on the conceptual elements in computer programming and practical experiences. Students are required to attend a number of laboratory sessions to practice what they have learned in the lectures. This practical work will help the students to reinforce the programming skills learned for applications.

**Teaching/Learning Methodology**

The emphasis of the course will be placed on learning the conceptual elements in computer programming. The teaching/learning methodology includes:

1. Lectures
2. Laboratory experiments
3. Tutorials
4. Assignments
5. Quizzes and Tests

**Assessment Methods in Alignment with Intended Learning Outcomes**

Specific assessment methods/tasks	% weighting	Intended subject learning outcomes to be assessed (Please tick as appropriate)					
		a	b	c	d	e	
1. Assignments	60%	✓	✓	✓	✓	✓	
2. Lab exercises		✓	✓	✓	✓	✓	
3. Quizzes and Tests		✓	✓		✓	✓	
4. Examination	40%	✓	✓		✓	✓	
Total	100 %						

For individual programming assignment, it can help the students to understand the subject contents, practice the programming skills, and develop the problem solving skills.

	<p>For <u>laboratory experiments participation</u>, it can help the students to practice and consolidate the concepts learned in the lectures.</p> <p>For <u>the quiz, test and exam assessment</u>, they are used to assist students to study the course in many ways, such as understanding the programming elements for solving computing problems, designing solutions for computing problems in a logical and systematic manner, and writing programs in a stepwise way.</p>	
<b>Student Study Effort Required</b>	Class contact:	
	▪ Lecture	28 Hrs.
	▪ Tutorial and lab exercises	14 Hrs.
	Other student study effort:	
	▪ Reading textbooks and course materials	84 Hrs.
	▪ Programming assignments	18 Hrs.
	Total student study effort	144 Hrs.
<b>Reading List and References</b>	<p><b>Textbooks:</b></p> <ol style="list-style-type: none"> <li>1. Y. Daniel Liang, Introduction to Java Programming, 6<sup>th</sup> edition, Prentice-Hall, 2007.</li> </ol> <p><b>Reference Books:</b></p> <ol style="list-style-type: none"> <li>1. Deitel &amp; Deitel, Java: How to Program, 7<sup>th</sup> edition, Prentice-Hall, 2007.</li> <li>2. John Lewis and William Loftus, Java Software Solutions, Foundations of Program Design, 5<sup>th</sup> edition, Prentice-Hall, 2007.</li> <li>3. Various materials on the web, especially <a href="http://java.sun.com">http://java.sun.com</a></li> </ol>	