City University of Hong Kong

Information on a Course
offered by Department of Computer Science
with effect from Semester A in 2012 / 2013

Part I

Course Title:  Introduction to Computer Programming

Course Code: CS1302

Course Duration: One Semester

No. of Credit Units: 3

Level: B1

Medium of Instruction: English

Prerequisites: (Course Code and Title)
Level 2 or above in HKDSE Information & Communication Technology with Software Development elective or
Pass in a placement test administered by CS Department

Precursors: (Course Code and Title)
Nil

Equivalent Courses: (Course Code and Title)
Nil

Exclusive Courses: (Course Code and Title)
CS1102 Introduction to Computer Studies

Part II

1. Course Aims:

This course aims to introduce to students with key concepts, techniques, and good practices of programming using a high-level programming language. Prior basic knowledge of computing and programming is assumed.
2. **Course Intended Learning Outcomes (CILOs)**

*Upon successful completion of this course, students should be able to:*

<table>
<thead>
<tr>
<th>No.</th>
<th>CILOs</th>
<th>Weighting (approx.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>explain the structure of a computer program;</td>
<td>10%</td>
</tr>
<tr>
<td>2.</td>
<td>analyze, test and debug computer programs;</td>
<td>20%</td>
</tr>
<tr>
<td>3.</td>
<td>apply proper programming techniques to solve a task;</td>
<td>50%</td>
</tr>
<tr>
<td>4.</td>
<td>construct well-structured programs.</td>
<td>20%</td>
</tr>
</tbody>
</table>

3. **Teaching and learning Activities (TLAs)**

*(designed to facilitate students’ achievement of the CILOs)*

Lectures – Various programming concepts and techniques will be introduced, explained and demonstrated with examples.

Tutorials – The tutorial sessions are designed to enable the students to put theory into practice and be proficient in a programming language. The exercises consist of programming tasks and students can try out their programs using the computer. Feedback will be given to students on their work.

Quizzes – The quizzes assess the students’ achievements at various stages within the semester. This provides formative assessment on their learning progress.

Assignments – The assignments are more challenging tasks compared with tutorials exercises. The students need to analyze the requirements and design programming solutions by applying (and combining) various techniques learnt from lectures and tutorials exercises. They are also required to construct their solutions as practical computer programs, and to explain their ideas/algorithms using suitable presentation methods (e.g. a report, flowchart, etc).

Suggested lecture/tutorial/laboratory mix: 2 hrs. lecture; 1 hr. tutorial.

<table>
<thead>
<tr>
<th>CILO No</th>
<th>Lectures</th>
<th>Tutorials</th>
<th>Assignments</th>
<th>Quizzes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Major</td>
<td>Major</td>
<td>--</td>
<td>Minor</td>
</tr>
<tr>
<td>2</td>
<td>Minor</td>
<td>Major</td>
<td>Major</td>
<td>Minor</td>
</tr>
<tr>
<td>3</td>
<td>Major</td>
<td>Major</td>
<td>Major</td>
<td>Major</td>
</tr>
<tr>
<td>4</td>
<td>Major</td>
<td>Major</td>
<td>Major</td>
<td>Minor</td>
</tr>
</tbody>
</table>

*(Major: Main focus on the CILO, Minor: Minor focus on the CILO)*
4. **Assessment Tasks/Activities**
*(designed to assess how well the students achieve the CILOs)*

Sample Assessment Tasks / Activities and Criteria of Each CILO:

<table>
<thead>
<tr>
<th>CILO No</th>
<th>Assessment criteria</th>
<th>Assessment tasks / activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Correctly explain different parts of a computer program and the behaviour of its execution.</td>
<td>Quizzes (Minor)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Exam (Minor)</td>
</tr>
<tr>
<td>2</td>
<td>Select proper test cases to assess the correctness of a program.</td>
<td>Assignment (Major)</td>
</tr>
<tr>
<td></td>
<td>Find out program errors and make corrections.</td>
<td>Quizzes (Minor)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Assignment (Major)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Exam (Minor)</td>
</tr>
<tr>
<td>3</td>
<td>Apply proper programming techniques to solve a task.</td>
<td>Quizzes (Major)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Assignment (Major)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Exam (Major)</td>
</tr>
<tr>
<td>4</td>
<td>Construct well-structured programs.</td>
<td>Quizzes (Minor)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Assignment (Major)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Exam (Minor)</td>
</tr>
</tbody>
</table>

5. **Grading of Student Achievement:** Refer to Grading of Courses in the Academic Regulations.

*Examination duration: 2 hours*

*Percentage of coursework, examination, etc.: 40% CW; 60% Exam*

*Grading pattern: Standard (A+AA--..F)*

For a student to pass the course, at least 30% of the maximum mark for the examination must be obtained.

**Part III**

Keyword Syllabus:

The development of algorithms, program design, programming language, control structures, data types, arrays, functions and parameters, composite data types, structured decomposition, programming style, program testing, introduction to recursion

Syllabus:

1. **Program development environment**

2. Programming techniques and the development of algorithms
   Modular decomposition and stepwise refinement, principles of abstraction.
   Algorithms, the realisation of algorithms as programs. Program design:
   programming language, procedural abstraction, parameter-passing, control
   structures, iteration, recursion.

3. Data structures
   The concept of data types. Simple data types. Arrays. Strings. Composite data
types.

4. Program development practice
   Elements of programming style. Program testing. Program documentation.

Recommended Reading:

**Essential Text**


**Supplementary Reading**
JavaScript for programmers, Paul J. Deitel, Harvey M. Deitel, Publisher Upper