

ATENEEO DE MANILA UNIVERSITY  
 Loyola Schools  
 Generic Course Syllabus for 2<sup>nd</sup> Semester, School Year 2012-2013

Department	Information Systems and Computer Science	School	Science and Engineering
------------	--	--------	-------------------------

Course No.	CS 21b
Course Title	Introduction to Computing II
No. of Units	3

Course Description:

This course builds on the basic programming concepts and techniques taught in CS 21a, and covers more advanced topics with the goal of teaching students how to write more complex and more powerful programs. Topics to be covered include inheritance and basic object-oriented design, graphics, multithreaded programming, file and streams, and basic networking.

Course Objective/s:

- To master object-oriented programming concepts
- To be able to design and write multi-user, multi-threaded, and networked applications in Java

Course Outline:

WEEK / TOPIC	LEARNING OBJECTIVES	ACTIVITIES	STUDENT OUTPUT
1 Introduction ♦ Diagnostic ♦ CS 21a review	♦ Review concepts like inheritance and exceptions from CS21a	♦ Lecture	♦ Submission of Diagnostic Test
2 Binary Representation	♦ Explain how numbers are represented in computers	♦ Exercise on Binary Representation	♦ Submission of exercise
3 Inheritance, Encapsulation and Access Modifiers ♦ Binary Representation	♦ Explain what encapsulation means ♦ Explain and	♦ Lecture ♦ Exercise on inheritance and ac-	♦ Submission of exercise

<ul style="list-style-type: none"> <li>◆ In-depth explanation of inheritance concepts</li> <li>◆ Access modifiers</li> </ul>	differentiate the different access modifiers	cess modifiers	
<p>4 Abstract Classes and Interfaces</p> <ul style="list-style-type: none"> <li>◆ Abstract classes</li> <li>◆ Interfaces</li> </ul>	<ul style="list-style-type: none"> <li>◆ Explain and implement abstract classes</li> <li>◆ Explain and implement interfaces</li> <li>◆ Explain when and how each is used.</li> </ul>	<ul style="list-style-type: none"> <li>◆ Lecture</li> <li>◆ Exercise on Abstract classes and interfaces</li> </ul>	<ul style="list-style-type: none"> <li>◆ Submission of exercise</li> </ul>
<p>5 Inner and Anonymous classes,</p> <ul style="list-style-type: none"> <li>◆ Inner classes and uses</li> <li>◆ Anonymous classes</li> <li>◆ Event handling</li> </ul>	<ul style="list-style-type: none"> <li>◆ Explain and implement inner and anonymous classes</li> <li>◆ Explain how to handle GUI events</li> <li>◆ Explain when and how each is used</li> </ul>	<ul style="list-style-type: none"> <li>◆ Lecture</li> <li>◆ Exercise</li> </ul>	<ul style="list-style-type: none"> <li>◆ Submission of exercise</li> </ul>
<p>6 Event Handling</p> <ul style="list-style-type: none"> <li>◆ Event listeners</li> </ul>	<ul style="list-style-type: none"> <li>◆ Explain the ActionListener interfaces</li> </ul>	<ul style="list-style-type: none"> <li>◆ Lecture</li> <li>◆ Exercise</li> </ul>	<ul style="list-style-type: none"> <li>◆ Submission of exercise</li> </ul>
<p>7 Graphics</p> <ul style="list-style-type: none"> <li>◆ Canvas and Graphics objects</li> </ul>	<ul style="list-style-type: none"> <li>◆ Draw low-level graphics on Frames/Applets</li> </ul>	<ul style="list-style-type: none"> <li>◆ Project 1 submission and defense</li> </ul>	<ul style="list-style-type: none"> <li>◆ Project submission and defense</li> </ul>
<p>8-9 Object-Oriented Design</p> <ul style="list-style-type: none"> <li>◆ UML Diagrams</li> </ul>	<ul style="list-style-type: none"> <li>◆ Design software using UML diagrams</li> </ul>	<ul style="list-style-type: none"> <li>◆ Lecture</li> <li>◆ Exercise on UML</li> </ul>	<ul style="list-style-type: none"> <li>◆ Submission of exercise</li> </ul>
<p>10 Midterm Exam</p>	<ul style="list-style-type: none"> <li>◆ Accomplish Midterm Assessment</li> </ul>	<ul style="list-style-type: none"> <li>◆ Midterm Exam</li> </ul>	<ul style="list-style-type: none"> <li>◆ Exam</li> </ul>
<p>11-12 Files, Streams, Strings, Exception Handling</p> <ul style="list-style-type: none"> <li>◆ Files and Streams</li> <li>◆ String manipulation</li> </ul>	<ul style="list-style-type: none"> <li>◆ Implement programs that open, read and write files</li> </ul>	<ul style="list-style-type: none"> <li>◆ Lecture</li> <li>◆ Exercise on Files and Streams</li> </ul>	<ul style="list-style-type: none"> <li>◆ Submission of exercise</li> </ul>

◆ Exception Handling	◆ Implement programs manipulate Strings		
13-16 Threads and Networking	◆ Implement programs that communicate across networks	◆ Lecture ◆ Exercises on Threads and Networking	◆ Submission of exercises
17 Finals ◆ Exam ◆ Project 2	◆ Accomplish Final Assessment	◆ Final Exam ◆ Project 2 submission and defense	◆ Exam ◆ Project submission and defense

References (optional):

Horstmann, Cay. Java Concepts, 6th ed. Wiley 2010  
 Barnes and Kolling. Objects First with Java: A Practical Introduction using BlueJ. Prentice Hall/Pearson Education, 2006  
 Savitch, Walter. Java: An Introduction to Computer Science & Programming. Pearson/Prentice-Hall, 2004  
 Schildt, Herbert. Java: A Beginner's Guide. McGraw Hill/Osborne, 2005  
 Wu, Thomas. An Introduction to Object-Oriented Programming with Java. McGraw-Hill Higher Education, 2004  
 Deitel, Deitel, and Santry, Advanced Java 2 Platform, Prentice Hall 2002

ATENEEO DE MANILA UNIVERSITY  
 Loyola Schools  
 Generic Course Syllabus for 2<sup>nd</sup> Semester, School Year 2012-2013

Department	Information Systems and Computer Science	School	Science and Engineering
------------	--	--------	-------------------------

Course No.	CS 105
Course Title	Theory of Algorithms
No. of Units	3

Course Description:

The objective of this course is to give students the basic tools necessary to develop their own algorithms, in whatever field of application they may be required. This course concentrates on the fundamental techniques used to design and analyze efficient algorithms. These techniques include greedy algorithms, divide-and-conquer algorithms, dynamic programming, and graph algorithms.

Course Objective/s:

To design algorithms for solving simple problems, and implement, test and debug these algorithms  
 To determine the time-complexity of different algorithms  
 To describe common applications for the simple data structures such as stacks, queues, trees and graphs  
 To identify the most appropriate data structures for different types of problems

Course Outline:

WEEK / TOPIC	LEARNING OBJECTIVES	ACTIVITIES	STUDENT OUTPUT
<b>1-2 Introduction</b> Why study data structure and algorithms? Number Representation	Differentiate among numeric systems Convert values across numeric systems Perform addition and subtraction in binary	Lecture Exercise on Number Representation	Submission of exercise
<b>3 Recursion</b> Definition Examples: Factorial, Power, and Array Reversal	Explain recursion in the context of iterations Implement recursive algorithms	Lecture Exercise on Recursion	Submission of exercise
<b>4 Searching</b> Unordered Se-	Explain and implement	Lecture Exercise on	Submission of exercise

<p>quence: Linear Search          Ordered Sequence:          Binary Search</p>	<p>search algorithms          Explain pros and cons of each implementation</p>	<p>Searching</p>	
<p><b>5-6 Sorting</b>          Bubble, Insertion and Selection Sort          Quick Sort          Merge Sort          Project</p>	<p>Explain and implement sorting algorithms          Explain pros and cons of each implementation</p>	<p>Lecture          Exercise on Sorting</p>	<p>Submission of exercise          Project submission and defense</p>
<p><b>7-8 Elementary Data Structures</b>          Review: Arrays vs. Linked Lists          Stacks          Queues</p>	<p>Explain and implement Stacks and Queues          Analyze implementations in terms of efficiency</p>	<p>Lecture          Exercise on Stacks/Queues</p>	<p>Submission of exercise</p>
<p><b>9 Midterms</b>          Exam</p>	<p>Accomplish Midterm Assessment</p>	<p>Midterm Exam          Project submission and defense</p>	<p>Exam</p>
<p><b>10-12 Trees</b>          General Trees          Binary Trees          Tree Implementation          Priority Queues          Dictionaries          Binary Search          Trees          Heaps</p>	<p>Explain and implement Trees and Heaps          Analyze implementations based on efficiency</p>	<p>Lecture          Exercise on Trees and Heaps</p>	<p>Submission of exercise</p>
<p><b>13 Advanced Sorting Algorithms</b>          Heap Sort          Bucket Sort          Radix Sort          Project</p>	<p>Explain and implement sorting algorithms          Explain pros and cons of each implementation</p>	<p>Lecture          Exercise on Sorting</p>	<p>Submission of exercise          Project submission and defense</p>
<p><b>14 Hashing</b></p>	<p>Explain and implement hashing</p>	<p>Lecture          Exercise on Hashing</p>	<p>Submission of exercise</p>
<p><b>15-16 Graphs</b>          Graph Implementations          Shortest Path          Minimum Cost          Spanning          Trees</p>	<p>Explain and implement the graph data structure          Explain and implement graph algorithms          Determine when to apply graph algorithms to problems</p>	<p>Lecture          Exercise on Graphs</p>	<p>Submission of exercise</p>

<b>17-18 Finals</b> Exam Project	Accomplish Final Assessment	Final Exam Project submission and defense	Exam Project submission and defense
--	--------------------------------	---	---

ATENEEO DE MANILA UNIVERSITY  
 Loyola Schools  
 Generic Course Syllabus for 2<sup>nd</sup> Semester, School Year 2012-2013

Department	Information Systems and Computer Science	School	Science and Engineering
------------	--	--------	-------------------------

Course No.	CS 112
Course Title	Structure of Programming Languages
No. of Units	3

Course Description:

Databases are now used extensively within organizations. Business applications of any significance will invariably use databases. Hence, it is necessary for the IT professional to understand both the technical as well as the managerial implications of database application development. This course aims to provide the basic database knowledge required of the professional.

Course Objective/s:

To understand what databases are and how they work;

- To be familiar with the basic theories, concepts, methods, and terminology used in database technology;
- To be acquainted with the necessary principles associated with the analysis, development, implementation, and maintenance of database systems; and
- To acquire database development tools, including entity-relationship diagrams, normalization process, SQL fundamentals, etc.

Course Outline:

WEEK / TOPIC	LEARNING OBJECTIVES	ACTIVITIES	STUDENT OUTPUT
1 Introduction ♦ Course Introduction ♦ Overview of Database Systems	♦ Define key terms ♦ Compare conventional file systems and database systems ü Identify typical components of a database system	♦ Lec-ture	

<p>2 Database Development Process</p> <ul style="list-style-type: none"> <li>◆ Database Development Life Cycle</li> </ul>	<ul style="list-style-type: none"> <li>◆ Describe the life cycle of a systems development project</li> <li>ü Explain the components and roles involved in the development process</li> </ul>	<ul style="list-style-type: none"> <li>◆ Lecture</li> </ul>	
<p>2-4 Data Modeling</p> <ul style="list-style-type: none"> <li>◆ Conceptual Data Model</li> <li>◆ Entities</li> <li>◆ Attributes</li> <li>◆ Relationships</li> <li>◆ Constraints</li> </ul>	<ul style="list-style-type: none"> <li>◆ Select appropriate names for entities, relationships, and attributes</li> <li>◆ Draw ERDs and EERDs to represent situations</li> </ul>	<ul style="list-style-type: none"> <li>◆ Lecture</li> <li>◆ Exercises on drawing ERDs</li> </ul>	<ul style="list-style-type: none"> <li>◆ Submission of exercises</li> </ul>
<p>TBA Project Deliverable 1</p>	<ul style="list-style-type: none"> <li>◆ Apply Data Modeling to the assigned business situation</li> </ul>	<ul style="list-style-type: none"> <li>◆ Group work</li> </ul>	<ul style="list-style-type: none"> <li>◆ Submission of deliverable</li> </ul>
<p>5-7 Logical Design</p> <ul style="list-style-type: none"> <li>ü Relational Database Model</li> <li>ü Keys</li> <li>ü Normalization</li> </ul>	<ul style="list-style-type: none"> <li>◆ Describe the relational model</li> <li>ü Select candidate keys</li> <li>ü Describe normal forms</li> <li>ü Transform ERDs to equivalent relations</li> </ul>	<ul style="list-style-type: none"> <li>◆ Lecture</li> <li>◆ Exercises on Normalization</li> </ul>	<ul style="list-style-type: none"> <li>◆ Submission of exercises</li> </ul>
<p>8 Midterm Exam</p>	<ul style="list-style-type: none"> <li>◆ Accomplish Midterm Assessment</li> </ul>	<ul style="list-style-type: none"> <li>◆ Midterm Exam</li> </ul>	<ul style="list-style-type: none"> <li>◆ Midterm Exam</li> </ul>
<p>9 Physical Design</p> <ul style="list-style-type: none"> <li>◆ Data Volume and Usage Analysis</li> <li>◆ Data Distribution Strategies</li> <li>◆ Indexes</li> <li>◆ File Organization</li> </ul>	<ul style="list-style-type: none"> <li>◆ Describe the physical database design process</li> <li>◆ Describe types of file organization</li> <li>◆ Explain the importance of indexes</li> </ul>	<ul style="list-style-type: none"> <li>◆ Lecture</li> <li>◆ Exercises on Volume and Usage Analysis</li> </ul>	<ul style="list-style-type: none"> <li>◆ Submission of exercises</li> </ul>



TBA Project Deliverable 2	<ul style="list-style-type: none"> <li>◆ Apply Logical and Physical Design to the assigned business situation</li> </ul>	<ul style="list-style-type: none"> <li>◆ Group work</li> </ul>	<ul style="list-style-type: none"> <li>◆ Submission of deliverable</li> </ul>
10-12 SQL <ul style="list-style-type: none"> <li>◆ Data Definition Language</li> <li>◆ Data Manipulation Language</li> <li>◆ Data Control Language</li> </ul>	<ul style="list-style-type: none"> <li>◆ Use SQL to define and manipulate a database</li> </ul>	<ul style="list-style-type: none"> <li>◆ Lecture</li> <li>◆ Hands-on exercises</li> </ul>	<ul style="list-style-type: none"> <li>◆ Submission of exercises</li> </ul>
13-14 Database Environments <ul style="list-style-type: none"> <li>◆ Client/Server</li> </ul>	<ul style="list-style-type: none"> <li>◆ Describe the client/server database environment</li> <li>◆ Explain linking tables using ODBC or JDBC</li> </ul>	<ul style="list-style-type: none"> <li>◆ Lecture</li> </ul>	
15 Administration <ul style="list-style-type: none"> <li>◆ Functions of Data and Database Administration</li> <li>◆ Security</li> <li>◆ Facilities and Techniques</li> </ul>	<ul style="list-style-type: none"> <li>◆ Describe the functions of data and database administration</li> <li>◆ Describe problems and techniques involved in database administration</li> </ul>	<ul style="list-style-type: none"> <li>◆ Lecture</li> </ul>	
16-17 Project Deliverable 3 and Presentations	<ul style="list-style-type: none"> <li>◆ Implement a prototype of the database system for the assigned business situation</li> <li>◆ Present accomplished work to a panel</li> </ul>	<ul style="list-style-type: none"> <li>◆ Group work</li> <li>◆ Presentation</li> </ul>	<ul style="list-style-type: none"> <li>◆ Submission of deliverable</li> <li>◆ Presentation</li> </ul>
18 Final Exam	<ul style="list-style-type: none"> <li>◆ Accomplish Final Assessment</li> </ul>	<ul style="list-style-type: none"> <li>◆ Final Exam</li> </ul>	<ul style="list-style-type: none"> <li>◆ Final Exam</li> </ul>

ATENEO DE MANILA UNIVERSITY  
Loyola Schools  
Generic Course Syllabus for 2<sup>nd</sup> Semester, School Year 2012-2013

Department	Information Systems and Computer Science	School	Science and Engineering
------------	--	--------	-------------------------

Course No.	CS 122
Course Title	Database Systems
No. of Units	3

Course Description:

Databases are now used extensively within organizations. Business applications of any significance will invariably use databases. Hence, it is necessary for the IT professional to understand both the technical as well as the managerial implications of database application development. This course aims to provide the basic database knowledge required of the professional.

Course Objective/s:

- To understand what databases are and how they work;
- To be familiar with the basic theories, concepts, methods, and terminology used in database technology;
- To be acquainted with the necessary principles associated with the analysis, development, implementation, and maintenance of database systems; and
- To acquire database development tools, including entity-relationship diagrams, normalization process, SQL fundamentals, etc.

Course Outline:

Introduction

- Course Introduction
- Overview of Database Systems

Database Development Process

- Database Development Life Cycle

Data Modeling

- Conceptual Data Model
- Entities
- Attributes
- Relationships Constraints

Logical Design

- Relational Database Model
- Keys Normalization

### Physical Design

- Data Volume and Usage Analysis
- Data Distribution Strategies
- Indexes
- File Organization

### SQL

- Data Definition Language
- Data Manipulation Language
- Data Control Language

### Database Environments

- Client/Server

### Administration

- Functions of Data and Database Administration
- Security
- Facilities and Techniques

### References (optional):

MySQL.com : <http://www.mysql.com>

ATENEO DE MANILA UNIVERSITY  
Loyola Schools  
Generic Course Syllabus for 2<sup>nd</sup> Semester, School Year 2012-2013

Department	Information Systems and Computer Science	School	Science and Engineering
------------	--	--------	-------------------------

Course No.	CS 162a & b
Course Title	Systems Programming
No. of Units	3

Course Description:

This course is an introduction to the techniques used to implement operating systems. It is also about the design and implementation of various popular operating systems like Windows and Linux. Among the topics covered will be process management, processor scheduling, deadlocks, main memory management, virtual memory management, control of disks and other input/output devices, file system structure and implementation, and protection and security.

Course Objective/s:

By the end of this course, students should be able to:

- Gain an appreciation of the operating systems that facilitate interaction with computers.
- Know the fundamental concepts behind operating systems and how they are applied to a variety of popular systems like Microsoft Windows, Unix, Mac O/S, and others.
- Create C or C++ programs that use Unix-specific libraries or features.
- Create programs that simulate several aspects of O/S behavior through the application of theories and concepts.