AP COMPUTER SCIENCE AB

SYLLABUS

Course Overview

AP Computer Science AB is taught over a period of one year, extending the concepts studied in the previous two courses. The first previous course, Computer Science I, covers introductory Java programming, methods, decisions, loops, input and output, using files, and using classes. The second previous course, Computer Science II, covers designing and writing classes, arrays, searching, sorting, and an introduction to Big-Oh. The language of all 3 courses is Java.

The AP Computer Science AB course emphasizes object-oriented programming and class design. Students show proficiency in creating and implementing classes, including the use of inheritance by extending classes and implementing interfaces. The Java Library classes are studied and used throughout the course. Encapsulation, abstraction, inheritance, and polymorphism are all studied. Another emphasis is the organization of information through the implementation of data structures. One and two-dimensional arrays, array lists, linked lists, stacks, queues, trees, sets, maps, and hash tables are all implemented and used. Algorithm analysis using Big-Oh notation and recursion are also studied. Students learn to analyze many large projects of interacting classes, including the Grid World Case Study. All students become familiar with the interaction of hardware and software components and the ethical and social implications of computing systems.

The content and objectives of this course include all the course objectives for AP Computer Science AB which are listed in the AP Computer Science Course Description. The course enhances the students’ problem solving ability and analytical skills, preparing them for the college computer science environment. All 6 topics in the course description are covered including object oriented program design, program implementation, program analysis, standard data structures, standard algorithms, and computing in context.

All AP Computer Science classes are taught in a modern computer lab. Lectures are very interactive; students take notes electronically, program the projects along with the teacher, print code and notes to add to a binder, and transfer programs to flash drives for work at home.
COMPUTER FACILITIES
Our lab contains 20 two year old PC’s running Windows XP with CD-RW drives, a data projector, and laser printers. Students have accounts on a networked server and store all of their class files in their accounts. The lab is open both before and after school, and students may stop in to use the lab during their study halls.

All computers include the Sun JDK 1.6 and JCreator 4.5 installed on C: drive for quick processing. All students receive a CD with the software so that they may install it on their home computer.

TEXTBOOKS AND RESOURCE MATERIALS
- AP Central Computer Science AB Quick Reference Guide for JDK 1.5
- Current magazine and Internet articles discussing ethical and social issues relating to computer use.
COURSE OUTLINE

Unit 1 (10 days)  REVIEW OF TOPICS FROM COMPUTER SCIENCE I AND II

AP Course Description Topics Covered:
1. Object Oriented Program Design
2. Program Implementation
3. Program Analysis
4. Standard Data Structures
5. Standard Algorithms
6. Computing in Context

Activities:
Review Java language features: import statements, library classes, comments, indentation and braces, primitive data types, methods, declaring local variables, arithmetic operators, boolean expressions, relational and boolean operators, if and select case, loops, strings and their methods, applets and graphics

Classes, encapsulation, and abstraction
Arrays
Enhanced for loops
Linear and binary searches
Selection Sort
Documentation, specifications, pre and post conditions
Hardware and software components of a computer system and how they interact
Networks and their operation
Computer ethics, piracy, and copyright laws

Assignments:
Write Car and Point class
Multiple Choice Questions
Free Response Question on simulating a deck of cards
Design questions from earlier year AP exams

Readings:
Unit 2 (7 days)  ALGORITHM ANALYSIS AND BIG-OH

AP Course Description Topics Covered:
   3. Program Analysis

Activities:
   Formula for summations
   Review of logarithms
   O(1), O(logN), O(N), O(N^2), O(N^3), O(NLogN), O(2^N)
   Best, worst, and average cases
   Space analysis
   Analyzing code and determining Big-Oh running time
   Comparing algorithms to solve problems efficiently
   Numerical representations and limits

Assignments:
   Multiple Choice and Free Response Questions from past years
   with an emphasis on Big-Oh analysis and comparisons

Readings:
   Big Java: Pages 712 – 740

Exam:
   6 Multiple Choice on Big-Oh and 2 Free Response
   (designing solution for a Gas Station class, analyzing stock and bond
   investments)

Unit 3 (14 days)  ARRAYLIST<E> AND ITS USE

AP Course Description Topics Covered:
   4. Standard Data Structures

Activities:
   List methods add, size, get, set
   Big-Oh of each method above
   Method remove and its Big-Oh
   Using Iterator<E> and its methods next, hasNext, and remove
   Autoboxing
   Practice multiple choice questions on above topics
   Compare and contrast all data structures studied so far in terms of which is best to
   solve a problem
Assignments:
   Design a Bank class using ArrayList<Bank>
   Design a class WordList and its methods
   Design a Concentration Board

Readings:
Big Java: Pages 284 – 298, 842 – 861

Exams:
  2 Free Response (2005 Student Grades, Number Set)
  20 Multiple choice questions

Unit 4 (5 days)   SORTING LISTS

AP Course Description Topics Covered:
   5. Standard Algorithms

Activities:
   Insertion sort
   Bubble sort
   Big-Oh analysis of each sort

Assignments:
   Multiple choice questions on sorting and searching
   Design a class Student, design a class StudentList, and sort the list

Readings:
Big Java: Pages 713 – 715, 731 – 740

Exams:
  Computer Science A Practice Exam on material covered so far

Unit 5 (12 days)   RECURSION

AP Course Description Topics Covered:
   2. Program Implementation
   5. Standard Algorithms

Activities:
   Writing recursive methods
   Tracing recursive methods
   Ordinary and tail recursion
   Quick sort
   Merge sort both iterative and recursive
Big-Oh analysis of above sorts
Recursion in applets and graphics

Assignments:
- Worksheet on 15 different recursive methods
- Multiple choice questions from earlier year’s exams involving recursion
- Recursive binary search
- Finding number of paths through a maze
- Towers of Hanoi
- Project on anagrams
- Project on address books
- I continue to assign recursive method projects during the rest of the course.

Readings:
Big Java: Pages 664 – 701, 715 – 725

Exams:
- 2 recursive methods exams
- Computer Science A Practice Exam on material covered so far

Unit 6 (15 days) TWO DIMENSIONAL ARRAYS

AP Course Description Topics Covered:
- 4. Standard Data Structures

Activities:
- Reading and printing
- Writing methods using two dimensional arrays
- Free response questions from previous years
- Compare and contrast all data structures studied so far in terms of which is best to solve a problem

Assignments:
- 1996 Free Response (SumCross and RemoveCross) changed into Java by me
- 1997 Free response (WordSearch) changed into Java by me
- 1998 Free Response (Black and White Pixels, recursive) changed into Java by me
- 2000 Free Response (Encryption) changed into Java by me
- 2001 Free Response (Window) changed into Java by me
- Multiple choice questions
- 1999 Free Response (Quilt) changed into Java by me
- Project “Kill the Blob”
- Project “Simple Maze”
- Project “Tougher Maze”
- Project “Magic Square Class”
Readings:
Big Java: Pages 298 – 302

Exams:
5 Multiple Choice and 1 Free Response Question (SumBorder and BorderSum)
1 Free Response Question (2002 Free Response (Flight) changed into Java by me)

Unit 7 (7 days)  INHERITANCE

AP Course Description Topics Covered:
1. Object Oriented Program Design

Activities:
Review of encapsulation and abstraction
Review of interfaces
Extending classes
Polymorphism
Early and late binding
Abstract classes

Assignments:
BankAccount classes
Square and Rectangle classes
Person and Student classes
Athlete, Runner, and Marathoner classes
Multiple choice questions
Design questions from previous years’ exams

Readings:
Big Java: Pages 410 – 441, 468 – 514, 608 – 662

Exams:
Multiple Choice Questions on Inheritance
3 Free response Questions (Sales/Transaction Classes, Bird Class, Pet Interface)

Unit 8
(20 days over the course of the entire year)  GRID WORLD CASE STUDY

AP Course Description Topics Covered:
1. Object Oriented Program Design
2. Program Implementation
3. Program Analysis
4. Standard Data Structures  
5. Standard Algorithms  
6. Computing in Context

Activities:
- Running the Case Study in JCreator  
- Experimenting with the simulation  
- Identifying the classes  
- Bug variations  
- Grid World classes and interfaces: Location, Grid, Actor, Rock, Flower, Bug  
- Interactions of objects: class Critter and its extensions  
- Grid Data Structures: AbstractGrid, BoundedGrid, UnboundedGrid  
- Changing the grid to a different data structure  
- Compare and contrast all data structures used in GridWorld in terms of which is best to solve a problem

Assignments:  
(from draft copy of GridWorld, will be changed to final copy when published)  
- Page 8 #1 – 4; Page 11 #1 – 7  
- Page 12 #1 – 5; Page 18 #1 – 5  
- Page 20 #1 – 4; Page 22 #1 – 5  
- Page 24 #1 – 11; Page 24 #1 – 4  
- Page 28 #1 – 6; Page 30 #1 – 6  
- Page 31 #1 – 7; Page 32 #1 – 6  
- Page 33 #1 – 4; Page 36 #1 – 5  
- Page 37 #1 – 8; Page 38 #1 – 5  
- Page 38 #1 – 3

Readings:  
*Grid World Case Study*: Pages 1-38  
Quick Reference Guide Appendices A-F

Exams:  
- Quiz Multiple Choice  
- Test Multiple Choice and Free Response  
- GridWorld multiple choice and free response questions will be included on other exams throughout the year.

**MIDTERM EXAM**  
40 Multiple Choice and 4 Free Response Questions on all material covered so far.  
This exam is given over 2 days with extended class periods.
Unit 9 (15 days) LINKED LISTS

AP Course Description Topics Covered:
4. Standard Data Structures
5. Standard Algorithms

Activities:
- How linked lists work
- Advantages of using them
- Writing our own singly linked list class
- Big-Oh of linked list methods
- Shallow vs. deep copies
- Writing our own doubly, circular, and doubly circular linked list classes and their methods
- List Iterator<E> and its methods add and set
- Writing our own methods iterator and listIterator for our linked list class
- The built in Java LinkedList<E> class and its methods
- Comparison of ArrayList<E> and LinkedList<E>
- Compare and contrast all data structures studied so far in terms of which is best to solve a problem

Assignments:
- Iterative and recursive methods for our own singly linked list class and its variations:
  - (addFirst, addLast, get, set, toString, getFirst, size, removeFirst, insertInOrder)
- Review assignment on Grid World (class design)
- Review assignment on two dimensional arrays (Bar Codes)
- 2003 Free Response (File Drawers)
- Methods to add or remove from a doubly, circular, or doubly circular linked list
- Write methods iterator, listIterator, next, hasNext, and remove for our classes

Readings:
Big Java: Pages 742 – 762

Exams:
- Practice Exam 40 Multiple Choice Questions
- 2 Free Response Questions on Linked Lists

Unit 10 (6 days) STACKS AND QUEUES

AP Course Description Topics Covered:
4. Standard Data Structures
5. Standard Algorithms
Activities:
  Implement Stack\(<E>\) as an array, ArrayList\(<E>\), and LinkedList\(<E>\)
  Infix, prefix, and postfix
  Implement Queue\(<E>\) as an array, ArrayList\(<E>\), and LinkedList\(<E>\)
  Compare and contrast all data structures studied so far in terms of which is best to solve a problem

Assignments:
  Various class implementations of a stack
  Project Palindrome
  Project “Matching Parentheses”
  Various class implementations of a queue
  Practice multiple choice questions on lists, stacks, and queues
  Project “Parking Garage”

Readings:
  *Big Java*: Pages 762 – 773

Exams:
  Free Response Question “Radix Sort”
  Free Response Question on stacks and queues

**Unit 11 (12 days) TREES**

AP Course Description Topics Covered:
  4. Standard Data Structures
  5. Standard Algorithms

Activities:
  Definitions and vocabulary
  Writing a Binary Tree abstract class and a Binary Search Tree class
  Using recursive methods
  Big-Oh for tree methods
  Compare and contrast all data structures studied so far in terms of which is best to solve a problem

Assignments:
  Tree methods search, preorder, inorder, postorder, and level by level traversals
  Tree statistic methods height, width, perimeter, nodeCount, isfull, isComplete
  Deleting a node
  97 AP Free Response (Separate) changed into Java by me
  Review project on Grid World
  Multiple choice questions on all topics covered so far
Readings:
*Big Java*: Pages 796 – 815

Exams:
- Multiple choice questions
- Free Response Tree questions

**Unit 12 (6 days)  HASH TABLES**

**AP Course Description Topics Covered:**
- 4. Standard Data Structures
- 5. Standard Algorithms
- 6.

**Activities:**
- Designing our own HashTable class
- The hashCode method
- Big-Oh analysis of hashTables
- Compare and contrast all data structures studied so far in terms of which is best to solve a problem

**Assignments:**
- Project (Hash Table searching)

**Readings:**
*Big Java*: Pages 783 – 795

**Exams:**
- Free Response Question using Linked Lists and Hash Tables

**Unit 13 (4 days)  SETS**

**AP Course Description Topics Covered:**
- 4. Standard Data Structures
- 5. Standard Algorithms

**Activities:**
- Set<E> methods
- TreeSet<E> and HashSet<E>: advantages of each, Big-Oh of each

**Assignments:**
- Project (Vocabulary list with no duplicates)
- Project (TreeSet<E> with Operations)
Readings:
*Big Java*: Pages 776 – 781

Exams:
- 25 Multiple Choice Questions

**Unit 14 (6 days)  MAPS**

AP Course Description Topics Covered:
- 4. Standard Data Structures
- 5. Standard Algorithms

Activities:
- Map\langle E, K \rangle \text{ methods}
- TreeMap\langle E, K \rangle \text{ and } HashMap\langle E, K \rangle : \text{ advantages of each, Big-Oh of each}
- Using Iterator\langle E \rangle \text{ with Sets and Maps}
- Compare and contrast all data structures studied so far in terms of which is best to solve a problem

Assignments:
- Project (Dictionary Map)

Readings:
*Big Java*: Pages 781 – 783

Exams:
- Grid World using Sets and Maps Free Response
- Acorn Book 15 Multiple Choice

**Unit 15 (8 days)  PRIORITY QUEUES AND HEAPS**

AP Course Description Topics Covered:
- 4. Standard Data Structures
- 5. Standard Algorithms

Activities:
- Definitions and vocabulary
- Big-Oh
- Designing and writing heap methods
- Heap sort
- Compare and contrast all data structures studied so far in terms of which is best to solve a problem
Assignments:
  - Write heap methods reheapUp and reheapDown
  - Code and test heap sort
  - Liver transplant project using priority queues
  - Min Heap Free Response question

Readings:
*Big Java*: Pages 815 – 839

Exams:
  - Multiple Choice questions on all covered topics
  - City/State Map

**Unit 16 (8 days throughout the year)**  
**COMPUTERS AND SOCIETY**

AP Course Description Topics Covered:
  7. Computing in Context

Activities:
Students will review processors, computer systems, and memory storage. They will investigate the use of different operating systems and compliers. They will continue to explore the interaction of hardware and software. The idea of intellectual property and copyright laws will be emphasized.

Students will also explore the effect of the computer on society. They will read and analyze articles from the media concerning careers, ethics, computer crimes, hardware and software, privacy, copyright laws and legal issues, and the responsible use of the computer.

Assignments:
  - Article summations
  - Oral presentations
  - Career investigations
  - Panel discussions

Readings:
  - Articles from the media and web sites

Exams:
  - Essays added to other exams
Unit 17 (10 days) EXAM REVIEW

AP Course Description Topics Covered:
1. Object Oriented Program Design
2. Program Implementation
3. Program Analysis
4. Standard Data Structures
5. Standard Algorithms
6. Computing in Context

Activities:
Go through 1 section of each type of question daily

Assignments:
2004 #1 (Library)
2004 #2 (Voter Ballots)
2004 #4 (Tree Priority Queue)
2005 #2 (Postal Codes)
2005 #3 (Tree Statistics)
2005 #4 (Email)
2004 40 Multiple Choice Questions
2006 #1 (Thesaurus)
2006 #2 (Product)
2006 #3 (WaitingList)
2007 Selected Free Response Questions
2 Free Response questions on Grid World

Readings:
Teacher handouts

Exams:
All students must take the AP Computer Science AB Exam.

Unit 18 (21 days)
This unit follows the A.P. exam and will cover various topics based on student needs and interests.

FINAL EXAM
40 Multiple Choice questions on all material covered during the course. This exam is given during an extended class period.
Correlation to AP Topic Outline – Computer Science AB

### I. Object-Oriented Program Design
The overall goal for designing a piece of software (a computer program) is to correctly solve the given problem. At the same time, this goal should encompass specifying and designing a program that is understandable, can be adapted to changing circumstances, and has the potential to be reused in whole or in part. The design process needs to be based on a thorough understanding of the problem to be solved.

<table>
<thead>
<tr>
<th>A. Program design</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Specify the purpose and goals for a problem.</td>
<td>Unit 1</td>
</tr>
<tr>
<td>2. Apply data abstraction and encapsulation.</td>
<td>Unit 1</td>
</tr>
<tr>
<td>3. Decompose a problem into classes; define relationships and responsibilities of those classes.</td>
<td>Unit 1</td>
</tr>
<tr>
<td>4. Understand and implement a given class hierarchy.</td>
<td>Unit 1</td>
</tr>
<tr>
<td>5. Identify reusable components from existing code using classes and class libraries.</td>
<td>Unit 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>B. Class design</th>
<th>Unit 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Design and implement a set of interacting classes.</td>
<td>Unit 1</td>
</tr>
<tr>
<td>2. Design an interface.</td>
<td>Unit 1</td>
</tr>
<tr>
<td>3. Choose appropriate advanced data structures and algorithms.</td>
<td>Unit 1</td>
</tr>
<tr>
<td>4. Apply functional decomposition.</td>
<td>All Units</td>
</tr>
<tr>
<td>5. Extend a given class using inheritance.</td>
<td>Unit 7</td>
</tr>
</tbody>
</table>

### II. Program Implementation
The overall goals of program implementation parallel those of program design. Classes that fill common needs should be built so that they can be reused easily in other programs. Object-oriented design is an important part of program implementation.

<table>
<thead>
<tr>
<th>A. Implementation techniques</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Methodology</td>
<td></td>
</tr>
<tr>
<td>a. Object-oriented development</td>
<td>Unit 1</td>
</tr>
<tr>
<td>b. Top-down development</td>
<td>Unit 1</td>
</tr>
<tr>
<td>c. Encapsulation and information hiding</td>
<td>Unit 1</td>
</tr>
<tr>
<td>d. Procedural abstraction</td>
<td>Unit 1</td>
</tr>
<tr>
<td>B. Programming constructs</td>
<td></td>
</tr>
<tr>
<td>1. Primitive types vs. objects</td>
<td>Unit 1</td>
</tr>
<tr>
<td>2. Declaration</td>
<td></td>
</tr>
<tr>
<td>a. Constant declarations</td>
<td>Unit 1</td>
</tr>
<tr>
<td>b. Variable declarations</td>
<td>Unit 1</td>
</tr>
<tr>
<td>c. Class declarations</td>
<td>Unit 1</td>
</tr>
<tr>
<td>d. Interface declarations</td>
<td>Unit 1</td>
</tr>
<tr>
<td>e. Method declarations</td>
<td>Unit 1</td>
</tr>
<tr>
<td>f. Parameter declarations</td>
<td>Unit 1</td>
</tr>
<tr>
<td>3. Console output (System.out.print/println)</td>
<td>Unit 1</td>
</tr>
<tr>
<td>4. Control</td>
<td></td>
</tr>
<tr>
<td>a. Methods</td>
<td>Unit 1</td>
</tr>
<tr>
<td>b. Sequential</td>
<td>Unit 1</td>
</tr>
<tr>
<td>c. Conditional</td>
<td>Unit 1</td>
</tr>
<tr>
<td>d. Iteration</td>
<td>Unit 1</td>
</tr>
<tr>
<td>e. Recursion</td>
<td>Unit 5</td>
</tr>
</tbody>
</table>

| C. Java library classes (included in the AB-level Java Subset) | All Units |
### III. Program Analysis

The analysis of programs includes examining and testing programs to determine whether they correctly meet their specifications. It also includes the analysis of programs or algorithms in order to understand their time and space requirements when applied to different data sets.

<table>
<thead>
<tr>
<th>A. Testing</th>
<th>All Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Test classes and libraries in isolation.</td>
<td>All Units</td>
</tr>
<tr>
<td>2. Identify boundary cases and generate appropriate test data.</td>
<td>All Units</td>
</tr>
<tr>
<td>3. Perform integration testing.</td>
<td>All Units</td>
</tr>
<tr>
<td>B. Debugging</td>
<td>All Units</td>
</tr>
<tr>
<td>1. Categorize errors: compile-time, run-time, logic.</td>
<td>All Units</td>
</tr>
<tr>
<td>2. Identify and correct errors.</td>
<td>All Units</td>
</tr>
<tr>
<td>3. Employ techniques such as using a debugger, adding extra output statements, or hand-tracing code.</td>
<td>All Units</td>
</tr>
<tr>
<td>C. Understand and modify existing code</td>
<td>All Units</td>
</tr>
<tr>
<td>D. Extend existing code using inheritance</td>
<td>Unit 7</td>
</tr>
<tr>
<td>E. Understand error handling</td>
<td>All Units</td>
</tr>
<tr>
<td>1. Understand runtime exceptions.</td>
<td>All Units</td>
</tr>
<tr>
<td>2. Throw runtime exceptions</td>
<td>All Units</td>
</tr>
<tr>
<td>F. Reason about programs</td>
<td>Unit 1</td>
</tr>
<tr>
<td>1. Pre- and post-conditions</td>
<td>Unit 1</td>
</tr>
<tr>
<td>2. Assertions</td>
<td>Unit 1</td>
</tr>
<tr>
<td>G. Analysis of algorithms</td>
<td>Unit 2</td>
</tr>
<tr>
<td>1. Informal comparisons of running times</td>
<td>Unit 2</td>
</tr>
<tr>
<td>2. Exact calculation of statement execution counts</td>
<td>Unit 2</td>
</tr>
<tr>
<td>3. Big-Oh notation</td>
<td>Unit 2</td>
</tr>
<tr>
<td>4. Worst-case and average-case time and space analysis</td>
<td>Unit 2</td>
</tr>
<tr>
<td>H. Numerical representations and limits</td>
<td>Unit 1</td>
</tr>
<tr>
<td>1. Representations of numbers in different bases</td>
<td>All Units</td>
</tr>
<tr>
<td>2. Limitations of finite representations (e.g., integer bounds, imprecision of floating-point representations, and round-off error)</td>
<td>All Units</td>
</tr>
</tbody>
</table>

### IV. Standard Data Structures

Data structures are used to represent information within a program. Abstraction is an important theme in the development and application of data structures.

| A. Simple data types (int, boolean, double)                              | Unit 1    |
| B. Classes                                                               | Unit 1    |
| C. One-dimensional arrays                                               | Unit 1 and 2 |
| D. Two-dimensional arrays                                               | Unit 6    |
| E. Linked lists (singly, doubly, circular)                               | Unit 9    |
| F. Stacks                                                                | Unit 10   |
| G. Queues                                                                | Unit 10   |
| H. Trees                                                                 | Unit 11   |
| I. Heaps                                                                 | Unit 15   |
| J. Priority queues                                                       | Unit 15   |
| K. Sets                                                                  | Unit 13   |
| L. Maps                                                                  | Unit 14   |

### V. Standard Algorithms

Standard algorithms serve as examples of good solutions to standard problems. Many are intertwined with standard data structures. These algorithms provide examples for analysis of program efficiency.

| A. Operations on AB-level data                                           | Unit 1,3,6,9 - 15 |
| 1. Traversals                                                           | Unit 1,3,6,9 - 15 |
| 2. Insertions                                                           | Unit 1,3,6,9 - 15 |
| 3. Deletions                                                            | Unit 1,3,6,9 - 15 |
| 4. Iterators                                                            | Unit 1,3,6,9 - 15 |
| B. Searching                                                            | Unit 1,3,6,9 - 15 |
1. Sequential  Unit 1  
2. Binary  Unit 1  
3. Hashing  Unit 12  
C. Sorting  Unit 1  
1. Selection  Unit 1  
2. Insertion  Unit 4  
3. Mergesort  Unit 5  
4. Quicksort  Unit 5  
5. Heapsort  Unit 15  

### VI. Computing in Context

A working knowledge of the major hardware and software components of computer systems is necessary for the study of computer science, as is the awareness of the ethical and social implications of computing systems. These topics need not be covered in detail but should be considered throughout the course.

<table>
<thead>
<tr>
<th>A. Major hardware components</th>
<th>All Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Primary and secondary memory</td>
<td>All Units</td>
</tr>
<tr>
<td>2. Processors</td>
<td>All Units</td>
</tr>
<tr>
<td>3. Peripherals</td>
<td>All Units</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>B. System software</th>
<th>All Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Language translators/compilers</td>
<td>All Units</td>
</tr>
<tr>
<td>2. Virtual machines</td>
<td>All Units</td>
</tr>
<tr>
<td>3. Operating systems</td>
<td>All Units</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C. Types of systems</th>
<th>All Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Single-user systems</td>
<td>All Units</td>
</tr>
<tr>
<td>2. Networks</td>
<td>All Units</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>D. Responsible use of computer systems</th>
<th>Unit 16</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. System reliability</td>
<td>Unit 16</td>
</tr>
<tr>
<td>2. Privacy</td>
<td>Unit 16</td>
</tr>
<tr>
<td>3. Legal issues and intellectual property</td>
<td>Unit 16</td>
</tr>
<tr>
<td>4. Social and ethical ramifications of computer use</td>
<td>Unit 16</td>
</tr>
</tbody>
</table>