COURSE DESCRIPTION

Dept., Number	CSci 111	Course Title	Computer Science I
Semester hours	3	Course Coordinator	Cynthia B. Zickos, Instructor

Current Catalog Description

Introduction to computer science with emphasis on problem solving and algorithm development. Using a high-level, block-structured programming language, students design, implement, debug, test, and document computer programs for various applications.

Textbook

Malik, D.S. *Java Programming: Program Design Including Data Structures*. Thompson Course Technology, Boston, Massachusetts, 2006. ISBN: 1–4188–3540–4.

References

Course Outcomes

This course is designed to provide the student with the opportunity to design, code, debug, document, and test computer program. Upon successful completion of the course, the students can:

1. explain the rudiments of program design and development;
2. use the basic components of a computer language, such as constants, variables, assignment statements, arithmetic and logical expressions, and console–based input and output;
3. use the selection statements: if, nested–if and switch;
4. use the repetition statements: while, for and do–while;
5. use array-processing techniques to solve a programming problem;
6. use simple algorithms, such as sentinel– or flag–controlled iteration.

Relationship between Course Outcomes and Program Outcomes

The course outcomes contribute to the program outcomes as shown below:

1. Explain the rudiments of program design and development: Outcomes b, c
2. Use the basic components of a computer language, such as constants, variables, assignment statements, arithmetic and logical expressions, and console–based input and output. Outcomes c, k
3. Use the selection statements: if, nested–if and switch: Outcomes a, c

4. Use the repetition statements: while, for and do–while: Outcomes a, c
5. Use array-processing techniques to solve a programming problem: Outcomes a, c, j
6. Use simple algorithms, such as sentinel- or flag-controlled iteration: Outcomes a, c

Prerequisites by Topic

Mathematical knowledge and skills equivalent to at least the successful completion of college algebra (Math 121 or higher Math course) or math ACT score of 21 or greater)

Major Topics Covered in the Course

1. Problem solving and the program development cycle
   a. Design process (structured and object–oriented)
   b. Tools for algorithm development (flowcharts, storyboards, or pseudocode)
   c. Implementation
   d. Testing and debugging
   e. Documentation
2. Programming methodologies
3. Reading and interpreting code
4. Programming standards and conventions
5. Basic programming concepts
   a. Intermediate versus executable code
   b. Compilation versus interpretation
   c. Primitive data types and type conversions
   d. Assignment statements
   e. Arithmetic expressions
   f. Logical expressions
   g. Memory allocation
   h. Declarations and scope
6. Control structures
   a. Sequence
   b. Selection
   c. Iteration
   d. Nesting
7. Array processing (one–dimensional)
   a. Populating
   b. Summing, averaging, and finding extrema
   c. Linear searching
8. Console input and output, formatted and unformatted
9. Classes
   a. Predefined classes and methods (e.g., Scanner, Math, and String)
   b. User–defined classes and methods (introduction)
   c. Methods and parameter passing
   d. Objects
   e. Reference variables

2

Assessment Plan for the Course

A comprehensive, 30-question exam constructed by a faculty committee is administered to each offering of CSCI 111. Student performance is analyzed question-by-question to identify needed adjustments in the textbook, lectures, or assignments. Faculty who regularly teach the class and the classes that follow (e.g., CSci 112, Computer Science II) participate in the evaluation, in the selection of textbooks, and in formulating a response appropriate to the assessment results.

How Data in the Course are Used to Assess Program Outcomes (unless adequately covered already in the assessment discussion under Criterion 4)

The standard exam administered in CSci 111 (see the previous item) is included in the curriculum-wide outcome assessment described in Chapters 2, 3, and 4 of the Self–Study.

Estimate Curriculum Category Content (Semester hours)

Area	Core	Advanced	Area	Core	Advanced
Algorithms	1		Software design	1	
Data structures			Concepts of programming languages	1	