

| | | | | | | | | | | |
|--|---|---|---------------------------|---|---|---|---|---|---|----|
| Dept Number | CS 202 | | Course Title | Introduction to Computer Science | | | | | | |
| Semester Hours | 4 | | Course Coordinator | Bill Cheng | | | | | | |
| | | | SP17 | | | | | | | |
| Catalog Description | An introduction to computers and programming using a high-level structured language including a discussion of programming constructs and data representation. Primary emphasis will be given to problem solving, algorithm design, and program development. The course meets for three lecture hours and two laboratory hours per week. | | | | | | | | | |
| Textbooks | | | | | | | | | | |
| SP17 | | | | | | | | | | |
| <i>Java from, control structures through objects: Gaddis, Tony PH Publications, 6th Edition , 2016 ISBN-978013395705-1</i> | | | | | | | | | | |
| References | | | | | | | | | | |
| Course Learning Outcomes | | | | | | | | | | |
| <ul style="list-style-type: none"> • To understand the fundamentals of computer hardware and software. • To learn programming and object-oriented design using Java. • To learn a disciplined and structured approach to the development of computerized solutions to problems. • To obtain a good foundation for further study in computer science. | | | | | | | | | | |
| Assessment of the Contribution to Student Outcomes | | | | | | | | | | |
| Outcome → | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Assessed → | X | X | X | X | | | | | | |
| Prerequisites by Topic | | | | | | | | | | |
| Mathematics 111 or equivalent with a grade of C or better. | | | | | | | | | | |

Major Topics Covered in the Course

1. Basic Concepts of Computer Systems
Computer organization and hardware: CPU, memory unit, I/O devices
Software: programs, operating systems, editors, compilers
Interacting with the operating system; using a screen editor; file system invoking the compiler
Computer systems: batch systems, interactive systems, mainframes, minicomputers, micros, networks
Programming languages: machine language, assembler language, high-level languages
Program Translation: source program, object program, compiler {2 classes}
2. Problem Solving Algorithms
Strategies: divide and conquer, special cases, generalization
Analysis: understanding the problem, specifying inputs and outputs
Pseudo code verification: hand checking, test data {3 classes}
3. Program Design and Development
Design methodologies: top-down, bottom-up, and combinations of the two, procedural abstraction, data abstraction, information hiding, object-oriented design
Structured programming techniques: use of appropriate control structure
Programming style: appropriate indentation, good identifier names
Documentation: appropriate commenting, self-documenting code
Testing and verification: bottom-up, top-down, debugging techniques {3 classes}
4. The Basics
Primitive data types; constants, variables and identifiers; named constants; arithmetic expressions; assignment statements {3 classes}
5. Input and Output
Console input and output, screen input and output, file input and output {3 classes}
6. Flow of Control
Conditions and logical expressions, relational operators, precedence rules
Conditional execution structures: if, if-else, switch
Iterative control structures: while, do-while, for
Nesting of control structures {6 classes}
7. Methods
Defining and calling methods; parameters; local variables; value returning methods and void methods; pre and post conditions {4 classes}
8. Arrays
Definition, processing, one-dimensional, two dimensional
Elementary searching and sorting {6 classes}
9. Strings {2 classes}
10. Classes and Objects
Constructors; instance variables and instance methods; static variables and static methods;
Overloading; instantiation of objects using the new operator; private and public; polymorphism and dynamic binding; inheritance and interfaces; encapsulation {6 classes}