

Brooklyn College
Department of Computer and Information Science

CISC 3120 [20.1] Design and Implementation of Software Applications 1

3 hours, 3 credits

Introduction to topics from diverse areas of computer and information science in an application-oriented context. Design, development, implementation and testing of a web-based, data-backed interactive application, such as an educational game or an e-commerce site. Human-computer interaction, graphics programming, net-centric computing, and software design. Learning code development by example, modification of instructor-authored code and independent code authoring. Application Programming Interfaces and state-of-the-art tools to design and partially prototype systems. Open-source technologies and their relationship to commercial technologies currently popular in workplaces. Application of fundamental computer skills and knowledge to rapidly changing job-specific technologies, tools and environments. Hands-on system development, supervised by the instructor. Students should be prepared to take CISC 3140 [20.2] in the following semester. It is recommended that students have taken or are taking CISC 3130 [22] in the semester they take CISC 3120 [20.1].

Objectives

Students will be able to:

1. Develop and test programs for real-world situations.
2. Integrate knowledge from a variety of sources to formulate a design for a multi-faceted, interactive, web-based computer system and implement it.
3. Analyze issues involving development and use of computing in modern society.
4. Demonstrate effective work on a team or in a working group.
5. Describe technical work orally and in writing.

Syllabus

Unit A. Interfaces

Week 1

Foundations of Human-Computer Interaction (HCI).

Weeks 2-3

Building a simple Graphical User Interface (GUI).

Week 4

Using APIs (Application Programmer Interfaces).

Unit B. Graphics

Weeks 5-6

Fundamental techniques in Graphics.

Week 7

Graphic systems.

Unit C. Net-centric Systems

Week 8

Introduction to Net-centric computing.

Week 9

Communication, Networking, and Network Security.

Weeks 10-11

The Web as client-server example.

Unit D. Software Design

Week 12

Introduction to Software Design.

Weeks 13-14

Software Requirements and Specifications.

Method of Assessment

1. Students will design and implement a graphical user interface and its underlying behavior that reacts to user input in a friendly way, including the handling of errors robustly.
2. Students will modify an animated graphics program. They will be provided with a working example, and they will have to design, implement and test a small modification to the program.
3. Students will modify an Internet-based client-server program. They will be provided with a working example, and they will have to design, implement and test a modification to the program that meets a set of given design criteria.
4. Students will design and implement a small database system. They will create queries that act on that database system, and they will implement those queries both from a command-line SQL interactive interface and from a web-based, browser-friendly database query language.
5. Students will modify an intelligent game-player. They will be provided with a working example of an automated game player and the game environment in which the player acts. They will be given goals for improving the performance of the player, and they will be required to design, implement, test and document a solution that meets the given goals.
6. Students will document their software design and methodologies for each program and/or project.
7. Students will design and implement a large final project in the second of the two courses, integrating concepts from multiple areas. They will be required to create documentation for the project. They will be required to fully test their project and submit a working program.

Bibliography

The Design of Everyday Things, by Donald Norman. BasicBooks (2002).

Designing the User Interface: Strategies for Effective Human-Computer Interaction (4th edition), by Ben Shneiderman and Catherine Plaisant. Addison Wesley (2004).

Introduction to Computer Graphics, by James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes, and Richard L. Phillips. Addison Wesley (1993).

The Java Tutorial: A Short Course on the Basics (4th edition), by Sharon Zakhour, Scott Hommel, Jacob Royal, Isaac Rabinovitch, Tom Risser and Mark Hoeber. Prentice Hall PTR (2006).

Programming with Java, by Barry Holmes. Jones and Bartlett Publishers (1998).

Computer Networking with Internet Protocols, by William Stallings. Prentice Hall (2004).

Understanding Unix/Linux Programming, by Bruce Molay. Pearson Education, Inc. (2003).

Core PHP Programming, by Leon Atkinson. Prentice Hall PTR (1999).

The Mythical Man-Month, by Frederick P. Brooks, Jr. Addison-Wesley Publishing Co, Inc., Anniversary Edition (1995).

Object-Oriented and Classical Software Engineering (7th edition), by Stephen R Schach. McGraw-Hill Science/Engineering/Math (2006).

An Introduction to Database Systems (8th Edition), by C.J. Date. Addison Wesley (2003).

Artificial Intelligence: A Modern Approach (2nd edition), by Stuart J. Russell and Peter Norvig. Prentice Hall (2002).