APPENDICES

APPENDIX A – COURSE SYLLABI

I. APCT 110/11

1. APCT 110 / 111 Introduction to Programming Lecture / Lab
2. Lecture: 2 credits and 2 contact hours / Lab: 1 credit and 1 contact hour
3. Coordinator: Dr. Briana Wellman
4. Textbook: Electronic interactive textbook: Programming in Python 3
   https://www.zybooks.com/catalog/programming-in-python-3
5. Specific course information
   a. Catalog description: Introduction to program development using a programming environment (i.e., Python). Topics covers a basic understanding of understanding of programming concepts and constructing numbers, strings, assignments, sequential vs. selective execution, nesting, loops, functions, arrays, reference parameters, file streams, etc.

   b. Prerequisites: None / Co-requisites: Lecture & Lab

   c. Required course
6. Specific goals for the course
   a. Students who complete this course should be able to perform the following tasks:

      i. Design programs using programming design tools.

      ii. Understand basic concepts such as data storage, program execution.

      iii. Demonstrate knowledge of fundamental programming concepts.

      iv. Understand the concept of object-oriented design.

      v. Demonstrate problem-solving skills

   b. Student Outcomes

      i. SO1-B Develop methods and algorithms to solve complex computing problems

ii. SO2-A Design a computing-based solution using appropriate design tools to meet a given set of requirements.

iii. SO5-A Participate as a team member or leader in developing and selecting ideas, establishing team goals and objectives, and creating a collaborative and inclusive environment

7. Brief list of topics to be covered
    a. Introduction to Python
    b. Variables and Expressions
    c. Types
    d. Branching
    e. Loops
    f. Functions
    g. Strings
    h. Lists and Dictionaries
    i. Classes
    j. Exceptions
    k. Modules
    l. Files
    m. Inheritance
    n. Recursion
    o. Plotting
    p. Searching and Sorting Algorithms

II. APCT 115

1. APCT 115: Computer Foundations

2. 3 credits and 3 contact hours

3. Coordinator: Dr. Anteneh Girma

4. Textbook: Brookshear, J. Glenn, “Computer Science, An Overview”, 12th ed., Prentice Hall, 2014

    a. other supplemental materials

        i. Reed, D., A Balanced Introduction to Computer Science, Prentice Hall, 3rd edition (September 10, 2010).

        ii. Dale, N. and Lewis, J., Computer Science Illuminated, Johnes & Bartlett Learning, 5th edition, 201

5. Specific course information

    a. Catalog description: Survey of computer science topics. Features applied concepts of iteration, induction, and recursion; functions and relations; propositional logic and predicate logic; graph and tree data structures; Boolean and computer logic; finite state machines; and algorithmic problem solving.

    b. Prerequisites: None / Co-requisites: None

    c. Required course

6. Specific goals for the course

    a. Students who complete this course should be able to perform the following tasks:

        i. Explain machine architecture including data storage and representation.

        ii. Explain the evolution of operating systems to the present day.

        iii. Explain networks and network security.

        iv. Understand the concept of an algorithm, Big-O analysis and software development.

        v. Understand data organization and data management systems.

        vi. Explain basic concepts of artificial intelligence and the theory of computation

        vii. Know the need for continued professional development and be able to specify ways to ensure this.

b. Student Outcomes

   i. SO3-B Deliver well-organized, logical oral presentations, including good explanations when questioned

   ii. SO4-A Demonstrate the knowledge of ACM Code of Ethics and Professional Conduct

   iii. SO4-B Recognize ethical and professional responsibilities of computing solutions and make informed judgments in computing practice based on legal and ethical principles

7. Brief list of topics to be covered

   a. Intro to computer hardware programs and history

   b. Data Representation, number systems and data manipulation

   c. Computer Organization

   d. Networks

   e. Operating systems

   f. Algorithms

   g. Program development including object design

   h. Arrays, searching and sorting

   i. Linked lists, stacks, queues, trees, binary trees

   j. Files structures and database systems

   k. Artificial Intelligence

   l. Theory of computation

III. APCT 231

1. APCT 231: Computer Science I Lecture

2. 3 credits and 3 contact hours

3. Coordinator: Dr. Lily Liang

4. Textbook: Introduction to Java Programming, Comprehensive Version 10th Edition, 9780133813463

   a. other supplemental materials

      i. Robert Sedgewick Kevin Wayne, Introduction to Programming with Java, 2nd Edition., Pearson (ISBN13:9780672337840)

5. Specific course information

      i. Catalog description: Covers algorithm and program development using a higher-level programming language (i.e., Java). Use of control structures, functions, and arrays. Objects are introduced.

      ii. Prerequisites: None / Co-requisites: APCT 233

      iii. Required course

6. Specific goals for the course

   a. Students who complete this course should be able to perform the following tasks:

      i. Be able to design problem solutions.

      ii. Be able to write, test and debug several computer programs independently and in peer groups.

      iii. Be able to understand programs written by others.

   b. Student Outcomes

      i. SO1-A Apply mathematical principles (algebra, calculus, and differential equation) and scientific principles to solve complex computing problems

      ii. SO1-B Develop methods and algorithms to solve complex computing problems

      iii. SO2-B Implement a computing-based solution to a computing problem to meet a given set of requirements

7. Brief list of topics to be covered

a. History /Background

b. IDE (Netbeans)

c. Java Structure

d. Selections

e. Mathematical Functions, Characters, and Strings

f. Loops

g. Methods

h. Single-Dimensional Arrays

i. Multidimensional Arrays

j. Objects and Classes

k. Object-Oriented Thinking

IV. APCT 233

1. APCT 233: Computer Science I Lab
2. 1 credit and 3 contact hours
3. Coordinator: Dr. Lily Liang
4. Textbook: Introduction to Java Programming, Comprehensive Version 10th Edition, 9780133813463
    a. other supplemental materials
        i. Robert Sedgewick Kevin Wayne, Introduction to Programming with Java, 2nd Edition., Pearson (ISBN13:9780672337840)
5. Specific course information
    a. Catalog description: Covers algorithm and program development using a higher-level programming language (i.e., Java). Use of control structures, functions, and arrays. Objects are introduced.
    b. Prerequisites: None / Co-requisites: APCT 231
    c. Required course
6. Specific goals for the course
    a. Students who complete this course should be able to perform the following tasks:
        i. Know how to use an IDE
        ii. Be able to write, test and debug several computer programs independently and in peer groups.
    b. Student Outcomes
        i. SO1-A Apply mathematical principles (algebra, calculus, and differential equation) and scientific principles to solve complex computing problems
        ii. SO1-B Develop methods and algorithms to solve complex computing problems
        iii. SO2-B Implement a computing-based solution to a computing problem to meet a given set of requirements
7. Brief list of topics to be covered
    a. Built-in data types

b. Standard I/O

c. Control structures

d. Selection

e. Looping

f. Functions

g. Classes and objects

h. Arrays

i. Techniques for testing and debugging

V. APCT 232

1. APCT 232: Computer Science II Lecture

2. 3 credits and 3 contact hours

3. Coordinator: Dr. Thabet Kacem

4. Textbook: Data Structures and Problem-Solving using Java, by Mark Allen Weiss, Fourth Edition, Addison-Wesley, ISBN: 9780321541406

    a. other supplemental materials

        i. Robert Sedgewick Kevin Wayne, Introduction to Programming with Java, 2nd Edition., Pearson (ISBN13:9780672337840)

5. Specific course information\

    a. Catalog description: Continuation of APCT 231. Emphasis on object-oriented programming (such as C++) Topics include multi-dimensional arrays, searching and sorting algorithms, data abstraction, file operations including random access files, classes, pointers, and introduction to linked lists, stacks and queues.

    b. Prerequisites: APCT 231/233 / Co-requisites: APCT 234

    c. Required course

6. Specific goals for the course

    a. Students who complete this course should be able to perform the following tasks:

        i. Use object-oriented programming techniques

        ii. Know how to implement sorting and searching algorithms

        iii. Use pointers, lists, stacks and queues in programs.

        iv. Design and code programs independently and in peer groups.

    b. Student Outcomes

        i. SO1-A Apply mathematical principles (algebra, calculus, and differential equation) and scientific principles to solve complex computing problems

        ii. SO1-B Develop methods and algorithms to solve complex computing problems

        iii. SO2-B Implement a computing-based solution to a computing problem to meet a given set of requirements.

7. Brief list of topics to be covered

    a. Oriented-object programming Concepts

        i. Classes

        ii. Objects

        iii. Inheritance

    b. Algorithm Analysis

    c. Searching and Sorting

    d. File Operations

    e. Recursion

    f. Stacks

    g. Lists

VI. APCT 234

1. APCT 234: Computer Science II Lab
2. 1 credit and 3 contact hours
3. Coordinator: Dr. Thabet Kacem
4. Textbook: Data Structures and Problem-Solving using Java, by Mark Allen Weiss, Fourth Edition, Addison-Wesley, ISBN: 9780321541406
   a. other supplemental materials
      i. Robert Sedgewick Kevin Wayne, Introduction to Programming with Java, 2nd Edition., Pearson (ISBN13:9780672337840)
5. Specific course information
   a. Catalog description: Continuation of APCT 231. Emphasis on object-oriented programming (such as C++) Topics include multi-dimensional arrays, searching and sorting algorithms, data abstraction, file operations including random access files, classes, pointers, and introduction to linked lists, stacks and queues.
   b. Prerequisites: APCT 231/233 / Co-requisites: APCT 232
   c. Required course
6. Specific goals for the course
   a. Students who complete this course should be able to perform the following tasks:
      i. Use object-oriented programming techniques
      ii. Know how to implement sorting and searching algorithms
      iii. Use pointers, lists, stacks and queues in programs.
      iv. Design and code programs independently and in peer groups.
   b. Student Outcomes
      i. SO1-A Apply mathematical principles (algebra, calculus, and differential equation) and scientific principles to solve complex computing problems
      ii. SO1-B Develop methods and algorithms to solve complex computing problems
      iii. SO2-B Implement a computing-based solution to a computing problem to meet a given set of requirements.

7. Brief list of topics to be covered
    a. Oriented-object programming Concepts
        i. Classes
        ii. Objects
        iii. Inheritance
    b. Algorithm Analysis
    c. Searching and Sorting
    d. File Operations
    e. Recursion
    f. Stacks
    g. Lists

VII. APCT 341

1. APCT 341: Advanced Web Development

2. 3 credits and 3 contact hours

3. Coordinator: Dr. Dong H Jeong

4. Textbook: PHP and MySQL Web Development, 4/E, Luke Welling, Laura Thomson, ISBN-10: 0672329166, 2009. Learning PHP, MySQL, JavaScript, and CSS: A Step-by-Step Guide to Creating Dynamic Websites (Second Edition) Author: Robin Nixon. ISBN-10: 1449319262

    a. other supplemental materials

        i. Web Style Guide, 3rd edition by Mr. Patrick J. Lynch and Ms. Sarah Horton.

        ii. PHP5 and MySQL Bible by Tim Converse, Joyce Park, Clark Morgan, May 7, 2004 | ISBN-10: 0764557467

        iii. PHP Programming Solutions, Vikram Vaswani, June 8, 2007 | ISBN-10: 007148745X

        iv. http://www.w3.org/TR/REC-html40/cover.html

        v. PHP Essentials, http://www.techotopia.com/index.php/PHP_Essentials

        vi. Video Lecture Materials, http://www.learnerstv.com/Free-Computer-Science-Video-lectures-ltv166-Page1.htm

5. Specific course information

    a. Catalog description: This course will focus on introducing advanced web programming languages such as PHP, JavaScript, ASP .Net, and ColdFusion. It mainly focuses on understanding advanced web-development techniques that use databases to create web contents.

    b. Prerequisites: CMOP 235, 236 Introduction to WebPage Development and HTM (Lec+Lab) / Co-requisites: None

    c. Elective course

6. Specific goals for the course

    a. Students who complete this course should be able to perform the following tasks:

        i. Understand advanced web programming concepts

        ii. Understand server-side web programming language

iii. Know how to build websites and post web contents on web server

iv. Know how to build a website as teamwork

b. Student Outcomes

i. SO 1-B: Develop methods and algorithms to solve complex computing problems.

ii. SO 2-B: Implement a computing-based solution to a computing problem to meet a given set of requirements.

iii. SO 6-B: Apply software development fundamentals to produce a computing-based solution.

7. Brief list of topics to be covered
   a. Storing and Retrieving Data using PHP
   b. Using Arrays in PHP
   c. String Manipulation and Regular Expressions in PHP
   d. Reusing Code and Writing Functions in PHP
   e. Object-Oriented PHP
   f. Error and Exception Handling in PHP
   g. Designing Your Web Database
   h. Creating Your Web Database
   i. Working with Your MySQL Database
   j. Accessing Your MySQL Database from the Web with PHP
   k. Advanced MySQL Administration
   l. Advanced MySQL Programming
   m. Running an E-commerce Site
   n. E-commerce Security Issues
   o. Web Application Security
   p. Implementing Authentication with PHP and MySQL
   q. Implementing Secure Transactions with PHP and MySQL
   r. Interacting with the File System and the Server
   s. Using Network and Protocol Functions
   t. Managing the Date and Time
   u. Generating Images
   v. Using Session Control in PHP
   w. Frameworks and Components
   x. Other Useful Features
   y. Using PHP and MySQL for Large Projects

VIII. CMOP 235

1. CMOP 235: Introduction to WebPage Development and HTML Lecture

2. 2 credits and 2 contact hours

3. Coordinator: Dr. Briana Wellman

4. Textbook: Patrick M. Carey, New Perspectives on HTML and CSS: Comprehensive, Cengage Learning; 7th edition, 2016, ISBN: 9781305578203

   a. other supplemental materials

      i. Online materials at http://www.w3schools

5. Specific course information

   a. Catalog description: This course in computer science develops basic skills in webpage development using the HTML programming language. It introduces the process of developing a webpage by explaining broadly known programming languages such as HTML, XHTML, CSS, and JavaScript.

   b. Prerequisites: APCT 231/233 / Co-requisites: CMOP 236

   c. Required course

6. Specific goals for the course

   a. Students who complete this course should be able to perform the following tasks:

      i. Use a variety of strategies and tools to create websites, independently and in peer group projects, to gain skills and experience needed for entry into web design and development careers.

      ii. Develop awareness and appreciation of the many ways that people access and utilize the web.

      iii. Understand the importance of the web as a medium of communication and societal concerns involved.

   b. Student Outcomes

      i. SO3-B Deliver well-organized, logical oral presentations, including good explanations when questioned

      ii. SO4-B Recognize ethical and professional responsibilities of computing solutions and make informed judgments in computing practice based on legal and ethical principles

    iii. SO5-B Ability to plan collaborative tasks, understand individual responsibility, share responsibilities and information on schedule, and engage in the success of team

7. Brief list of topics to be covered

    a. History of the World Wide Web and HTML

    b. Structure of HTML Document

    c. Elements and Attributes

    d. Inserting Inline Images

    e. Character Sets and Special Characters

    f. Page Layouts with CSS

    g. Lists

    h. Tables and Columns

    i. Working with Metadata

    j. Web Site Structures and Navigation

    k. Hypertext Links, Linked Images, and Image Maps

    l. Style Sheets with CSS for Color, Lists, Text Style, Display Style, Margins and Padding

    m. Web Forms

    n. Introduction to Multimedia in Web Pages

    o. Introduction to JavaScript

IX. CMOP 236

1. CMOP 236: Introduction to WebPage Development and HTML Lab
2. 1 credit and 1 contact hour
3. Coordinator: Dr. Briana Wellman
4. Textbook: Patrick M. Carey, New Perspectives on HTML and CSS: Comprehensive, Cengage Learning; 7th edition, 2016, ISBN: 9781305578203
    a. other supplemental materials
        i. Online materials at http://www.w3schools
5. Specific course information
    a. Catalog description: This course in computer science develops basic skills in webpage development using the HTML programming language. It introduces the process of developing a webpage by explaining broadly known programming languages such as HTML, XHTML, CSS, and JavaScript.
    b. Prerequisites: APCT 231/233 / Co-requisites: CMOP 235
    c. Required course
6. Specific goals for the course
    a. Students who complete this course should be able to perform the following tasks:
        i. Use a variety of strategies and tools to create websites, independently and in peer group projects, to gain skills and experience needed for entry into web design and development careers.
        ii. Develop awareness and appreciation of the many ways that people access and utilize the web.
        iii. Understand the importance of the web as a medium of communication and societal concerns involved.
    b. Student Outcomes
        i. SO3-B Deliver well-organized, logical oral presentations, including good explanations when questioned
        ii. SO4-B Recognize ethical and professional responsibilities of computing solutions and make informed judgments in computing practice based on legal and ethical principles

iii. SO5-B Ability to plan collaborative tasks, understand individual responsibility, share responsibilities and information on schedule, and engage in the success of team

7. Brief list of topics to be covered

   a. History of the World Wide Web and HTML

   b. Structure of HTML Document

   c. Elements and Attributes

   d. Inserting Inline Images

   e. Character Sets and Special Characters

   f. Page Layouts with CSS

   g. Lists

   h. Tables and Columns

   i. Working with Metadata

   j. Web Site Structures and Navigation

   k. Hypertext Links, Linked Images, and Image Maps

   l. Style Sheets with CSS for Color, Lists, Text Style, Display Style, Margins and Padding

   m. Web Forms

   n. Introduction to Multimedia in Web Pages

   o. Introduction to JavaScript

X. CSCI 241

1. CSCI 241: Data Structures
2. 3 credits and 3 contact hours
3. Coordinator: Dr. Thabet Kacem
4. Textbook: Adam Drozdek, Data Structures and Algorithms in C++ / Java, Course Technology; 4th edition, 2012
    a. other supplemental materials
        i. Savitch, W., Main, M., "Data Structures and Other Objects Using C++ / Java", Prentice Hall; 4 edition (March 6, 2010)
        ii. Goodrich, M., Tamassia, M., Data Structures and Algorithms in C++ / Java, Wiley; 5 edition (February 9, 2010)
5. Specific course information
    a. Catalog description: This course covers the design and implementation of data structures including arrays, stacks, queues, linked lists, binary trees, heaps, balanced trees and graphs. Other topics include sorting, hashing, memory allocation, and garbage collection.
    b. Prerequisites: APCT 232, 234 / Co-requisites: None
    c. Required course
6. Specific goals for the course
    a. Students who complete this course should be able to perform the following tasks:
        i. Further development of programming skills
        ii. The ability to analyze algorithms using big-O notation
        iii. The ability to implement and use elementary and advanced data structures in programs
        iv. The ability to understand and implement algorithms using recursion
    b. Student Outcomes
        i. SO1-C Estimate time and space complexity of algorithms
        ii. SO2-B Implement a computing-based solution to a computing problem to meet a given set of requirements

iii. SO6-A Apply computer science theory, principles and practices learned in various courses to produce a computing-based solution

7. Brief list of topics to be covered

   a. Review of Search and Sort Algorithms

   b. Algorithm Analysis and Big-O Notation

   c. Review of Linked Lists

   d. Stacks

   e. Queues

   f. Recursion

   g. Binary Trees

   h. Graphs

XI. CSCI 306

1. CSCI 306: Computer Ethics and Law

2. 3 credits and 3 contact hours

3. Coordinator: Dr. Briana Wellman

4. Textbook: A Gift of Fire: Social, Legal, and Ethical Issues for Computing, Technology, 5/E, Baase ISBN-10: 0134615271, ISBN-13: 9780134615271

   a. other supplemental materials: none

5. Specific course information

   a. Catalog description: A survey course that reviews implications and impacts of computing technology throughout the world. The course examines the policies that relate to the use of computer technology, such as privacy and national security, shared data and information, copyright and intellectual property, legislative and constitutional issues, changing labor force composition, and professional ethics.

   b. Prerequisites: Junior standing or above / Co-requisites: None

   c. Required course

6. Specific goals for the course

   a. Students who complete this course should be able to perform the following tasks:

      i. Identify implications and impacts of computer technology in global, economic, environmental and societal contexts

      ii. Identify the ACM/IEEE Code of Ethics and how it relates to professional and social responsibilities

      iii. Develop skills of critical analysis and applying ethical principles to situations and dialectical thinking

      iv. Communicate effectively through writing and interpersonal and group interaction

   b. Student Outcomes

      i. SO3-A Produce a variety of written documents using appropriate formats and grammar with discipline-specific conventions including citations appropriate to the audience

      ii. SO4-A Demonstrate the knowledge of ACM Code of Ethics and Professional Conduct

iii. SO4-B Recognize ethical and professional responsibilities of computing solutions and make informed judgments in computing practice based on legal and ethical

7. Brief list of topics to be covered

   a. Protecting Privacy

   b. Freedom of Speech

   c. Anonymity

   d. Intellectual Property

   e. Responses to Copyright Infringement

   f. Search Engines and Online Libraries

   g. Open Source Software

   h. Crime and Security (Hacking)

   i. Work-Impacts on Employment

   j. Evaluating and Controlling Technology

   k. Errors, Failures and Risks

   l. Professional Ethics Code and Responsibilities

XII. CSCI 325

1. CSCI 325: Organization of Programming Language

2. 3 credits and 3 contact hours

3. Coordinator: Dr. Byunngu Yu

4. Textbook: Sebesta, R., Concepts of Programming Languages 11th ed., Addison-Wesley, 2016

   a. other supplemental materials

      i. Tucker, A. B., Noonan, R. E., Programming Languages, Principles and Paradigms 2nd ed., McGraw-Hill, 2006

5. Specific course information

   a. Catalog description: The study of the organization of programming languages, especially the run-time behavior of programs; formal study of programming languages specification and analysis; continuation of the development of problem analysis and solution, and programming skills.

   b. Prerequisites: CSCI 241 / Co-requisites: None

   c. Required course

6. Specific goals for the course

   a. Students who complete this course should be able to perform the following tasks:

      i. To understand the run-time behavior of programs

      ii. To understand the major programming paradigms and how they have changed over time, coupled with the development of new programming languages.

      iii. To understand formal syntax specifications for programming languages

   b. Student Outcomes

      i. SO1-C Estimate time and space complexity of algorithms

      ii. SO2-B Implement a computing-based solution to a computing problem to meet a given set of requirements

      iii. SO6-A Apply computer science theory, principles and practices learned in various courses to produce a computing-based solution

7. Brief list of topics to be covered

a. Major paradigms of programming languages

b. History and evolution

c. Syntax and semantics

d. Lexical and syntax analysis

e. Names, binding, type checking, scope

f. Data types

g. Expressions and assignment statements

h. Statement-level control statements

i. Subprograms and implementation

j. Abstract data types

k. Object oriented programming (limited)

l. Concurrency

m. Exception Handling

n. Functional Programming

o. Logic Programming

XIII. CSCI 341

1. CSCI 341: Software Engineering

2. 3 credits and 3 contact hours

3. Coordinator: Dr. Junwhan Kim

4. Textbook: Software Engineering, 10th Edition, Ian Sommerville, ISBN-13: 978-0133943030, ISBN-10: 0133943038

   a. other supplemental materials: none

5. Specific course information

   a. Catalog description: Explores issues in design, development, documentation, coding and implementation of large software projects. The tools and techniques required for all stages are addressed. The functional requirements and decomposition of model problems are discussed. Validation, test and maintenance of large software systems are also covered.

   b. Prerequisites: APCT 232, 234 / Co-requisites: None

   c. Required course

6. Specific goals for the course

   a. Students who complete this course should be able to perform the following tasks:

      i. Understand the principles, concepts and techniques of Software Engineering

      ii. Develop software systems with an emphasis on applying sound, disciplined software engineering practice.

      iii. Work in a team environment and present group projects.

   b. Student Outcomes

      i. SO3-A Produce a variety of written documents using appropriate formats and grammar with discipline-specific conventions including citations appropriate to the audience

      ii. SO3-C Produce appropriate graphics such as figures, tables in written and oral communications

      iii. SO5-A Demonstrate ability to participate as a team member or leader in developing and selecting ideas, establishing team goals and objectives, and creating a collaborative and inclusive environment.

iv. SO5-B Ability to plan collaborative tasks, understand individual responsibility, share responsibilities and information on schedule, and engage in the success of team goals.

v. SO6-B Apply software development fundamentals to produce a computing-based

7. Brief list of topics to be covered

a. Overview

b. Requirements

c. Design

d. Development

e. Verification and validation

f. Managing people

g. Tests

XIV. CSCI 345

1. CSCI 345: Human Computer Interaction

2. 3 credits and 3 contact hours

3. Coordinator: Dr. Dong H Jeong

4. Textbook Designing The User Interface by B. Sheneiderman & C. Plaisant, Publisher: Addison Wesley; 5 edition (March 8, 2009) ISBN-10: 0321537351

    a. other supplemental materials: none

5. Specific course information

    a. Catalog description: This course provides an introduction to the field of human-computer interaction (HCI) that concentrates on the study of interaction between human (users) and computers. From this course, students learn a body of knowledge and a practical set of well known, tested, and necessary skills related to HCI.

    b. Prerequisites: Junior Standing / Co-requisites: None

    c. Elective course

6. Specific goals for the course

    a. Students who complete this course should be able to perform the following tasks:

        i. Understand the usability of interactive systems and ethical considerations of them

        ii. Understand how to manage design processes and how to evaluate interface designs

        iii. Understand interaction styles in different environments (virtual environment, GUI-based windows environment, etc.)

        iv. Know the various design issues (quality of service, balancing function and fashion, etc.)

    b. Student Outcomes

        i. SO2- B Implement a computing-based solution to a computing problem to meet a given set of requirements

        ii. SO3-B Deliver well-organized, logical oral presentations, including good explanations when questioned

        iii. SO4-B Recognize ethical and professional responsibilities of computing solutions that have impact in global, economic, environmental and societal contexts.

7. Brief list of topics to be covered

    a. Usability of interactive systems

    b. Guidelines, Principles, and Theories

    c. Managing design processes

    d. Evaluating interface designs

    e. Direct Manipulation and Virtual Environments

    f. Command and Natural Languages

    g. Interaction Devices

    h. Collaboration and Social Media Participation

    i. Quality of Service

    j. Balancing Function and Fashion

    k. User Documentation and Online Help

    l. Information Search

    m. Information Visualization

XV. CSCI 351

1. CSCI 351: Computer Networks

2. 3 credits and 3 contact hours

3. Coordinator: Dr. Junwhan Kim

4. Textbook: Computer Networks by Andrew S. Tanenbaum, Publisher: Prentice Hall; 5 edition (2012), ISBN-10: 0132126958

   a. other supplemental materials

      i. Computer Networking: A Top-Down Approach (7th Edition), by James Kurose, Keith Ross, ISBN-10: 9780133594140

5. Specific course information

   a. Catalog description: This course aims to provide data communication fundamentals and the principles governing computer communication networks. It provides an understanding of the components of networks, how they are connected as well as the basics in the design and implementation of network protocols. A number of techniques and protocols with respect to addressing, subnetting, routing, multicasting, and the interconnection of heterogeneous networks are discussed.

   b. Prerequisites: CSCI 241 / Co-requisites: None

   c. Required course

6. Specific goals for the course

   a. Students who complete this course should be able to perform the following tasks:

      i. Know the basic history, definitions, and terminology in computer networking

      ii. Learn fundamental principles and basic concepts of data communications and networks

      iii. Understand basic design and implementation of network protocols and security concerns

      iv. Understand the various components of a network and how they are connected

   b. Student Outcomes

      i. SO2-A Design a computing-based solution using appropriate design tools to meet a given set of requirements

ii. SO2-C Evaluate a computing-based solution by defining metrics and measuring the performance of the solution that meet a given set of requirements

iii. SO3-C Produce appropriate graphics such as figures, tables in written and oral communications

iv. SO6-C Identify risks of computing-based solutions and describe approaches to manage them

7. Brief list of topics to be covered

   a. Fundamental terminologies and definitions

   b. Reference Models

   c. Physical (transmission media, encoding) and MAC

   d. Data Link and Network (routing protocols and IP addressing)

   e. Transport Layer (TCP and UDP)

   f. Local Area Network (LAN) technologies

   g. Wireless network technologies

   h. Internetworking, IPv6, DNS, and SNMP

   i. Application Layer (WWW, Email, Multimedia)

XVI. CSCI 352

1. CSCI 352: Network Security
2. 3 credits and 3 contact hours
3. Coordinator: Dr. Anteneh Girma
4. Textbook: Computer Security Principles and Practice, 4th Edition, Publisher Pearson Edition 2018, ISBN-10:0-13-479410-9
    a. other supplemental materials
        i. Computer Security Principles and Practice, 4th Edition, Publisher Pearson Edition 2018, ISBN-10:0-13-479410-9
5. Specific course information
    a. Catalog description: This course provides a comprehensive overview of fundamental network security concepts, techniques, and issues such as types of attacks computers/networks are vulnerable to, attacker profiles, and hardware/software defense solutions available.
    b. Prerequisites: CSCI 241 / Co-requisites: None
    c. Elective course
6. Specific goals for the course
    a. Students who complete this course should be able to perform the following tasks:
        i. Learn fundamental principles and basic concepts of data communications and networking
        ii. Understand the various components of a network and how they are connected.
        iii. Describe the Different Network Layers (OSI and TCP/ IP Layers)
        iv. Understand and describe different remote-access network security protocols and their procedures.
        v. Identify the goals of network security, and what sorts of attacks do you need to defend against.
        vi. Describe the key security requirements of confidentiality, integrity, and availability.
        vii. Discuss the types of security threats and attacks that must be dealt with and identify the best security practices that can be implemented to protect organizational resources.

viii. Explain the fundamental security design principles.

ix. Discuss the use of attack surfaces and attack trees.

x. Understand the principle aspects of a comprehensive security strategy.

xi. Summarize the functional requirements for computer security

xii. Understand the difference between Intrusion prevention and intrusion detection systems, and how they do protect an organization from common security threats.

b. Student Outcomes

i. SO3-A Produce a variety of written documents using appropriate formats and grammar with discipline-specific conventions including citations appropriate to the audience

ii. SO4-B Recognize ethical and professional responsibilities of computing solutions and make informed judgments in computing practice based on legal and ethical principles.

iii. SO6-C Identify risks (security, safety, market, financial, technology, people, quality, structure and process) of computing-based solutions and describe approaches to manage them

7. Brief list of topics to be covered

a. Network Technologies

b. Network Layers

c. Network Security Goals

d. Categories of Network Attacks

e. Defending Against Network Attacks

f. Remote Access Security

g. Firewalls

h. Virtual Private Network

i. Intrusion Detection and Prevention (IDS/IPS)

XVII. CSCI 353

1. CSCI 353: Information Security

2. 3 credits and 3 contact hours

3. Coordinator: Dr. Dong Hyun Jeong

4. Textbook: Principles of Information Security, By: Michael E. Whitman; Herbert J. Mattord, Publisher: Cengage Learning, Print ISBN: 9781337102063, 1337102067

   a. other supplemental materials: none

5. Specific course information

   a. Catalog description: This course provides an in-depth understanding of general information security fundamentals, organization and operation security procedures and policies, handling of security incidents, security audit principles and practices, security ethics, and computer forensics.

   b. Prerequisites: junior standing or above / Co-requisites: None

   c. Elective course

6. Specific goals for the course

   a. Students who complete this course should be able to perform the following tasks:

      i. understanding the role of information security in computing

      ii. maintaining information or data securely

      iii. finding any possible vulnerability scenarios in computing systems

      iv. understanding secure procedures to protect information.

   b. Student Outcomes

      i. SO3-A Produce a variety of written documents using appropriate formats and grammar with discipline-specific conventions including citations appropriate to the audience.

      ii. SO4-B Recognize ethical and professional responsibilities of computing solutions and make informed judgments in computing practice based on legal and ethical principles

      iii. SO6-C Identify risks of computing-based solutions and describe approaches to manage them

7. Brief list of topics to be covered

a. Information Security
    i. History of information security
    ii. CNSS security model
b. Security
    i. Threats, Attacks, Secure software development
c. An overview of legal, ethical, and professional issues in information security
d. Risk identification and management
e. Policy and security planning
f. Physical security with access controls
g. Intrusion detection and prevention
h. Secure communication protocols
i. Security policies and security policy design

XVIII. CSCI 398

1. CSCI 398: Advanced Applied Programming
2. 3 credits and 3 contact hours
3. Coordinator: Dr. Timothy Oladunni
4. Textbook: Changes from semester to semester, dependent on topic selected
   a. other supplemental materials: none
5. Specific course information
   a. Catalog description: The course explores developing applied applications for various computing environments. This course will cover programming language(s) that is not covered in the CS I and CS II.
   b. Prerequisites: APCT 232, 234 / Co-requisites: None
   c. Elective course
6. Specific goals for the course
   a. Students who complete this course should be able to perform the following tasks:
      i. Develop, test and debug applied applications independently
      ii. Understand applied programming language
      iii. Understand how to design and create programs with the covered programming language
   b. Student Outcomes
      i. SO1-B Develop methods and algorithms to solve computing problems
      ii. SO2-B Implement a computing-based solution to a computing problem to meet a given set of requirements
      iii. SO6-B Apply software development fundamentals to produce a computing-based solution
7. Brief list of topics to be covered
   a. Intro to applied programming language
   b. Strings and numbers
   c. Control structures
   d. Files and processes

e. Lists, Arrays, and more

XIX. CSCI 410

1. CSCI 410: Theory of Computing
2. 3 credits and 3 contact hours
3. Coordinator: Dr. Thabet Kacem
4. Textbook: Daniel Cohen, Introduction to Computer Theory, 2nd ed, Wiley, 1996
    a. other supplemental materials
        i. Tucker, A. B., Noonan, R. E., Programming Languages, Principles and Paradigms 2nd ed., McGraw-Hill, 2007
5. Specific course information
    a. Catalog description: Introduction to the theory of computing including: Regular languages, finite automata, transition graphs, Kleene's theorem. Finite automata with output. Context-free languages, derivation trees, normal form grammars, pumping lemma, pushdown automata, and Turing machines.
    b. Prerequisites: CSCI 241, MATH 152 or 156, MATH 213 / Co-requisites: None
    c. Required course
6. Specific goals for the course
    a. Students who complete this course should be able to perform the following tasks:
        i. Understand and be able to solve problems using abstract mathematical machines
        ii. Master regular languages and finite automata, context-free languages, pushdown automata, and Turing recognizable languages
        iii. Have a broad overview of the theoretical foundations of computer science
        iv. Be familiar with thinking analytically in related areas of theory in computer science
    b. Student Outcomes
        i. SO1-A Apply mathematical principles (algebra, calculus and differential equation) and scientific principles to solve computing problems
        ii. SO1-C Estimate time and space complexity of algorithms
        iii. SO2-A Design a computing-based solution using appropriate design tools to meet a given set of requirements

7. Brief list of topics to be covered

   a. Languages and recursive definitions

   b. Regular Expressions

   c. Finite Automata

   d. Transition Graphs and Finite Automata with Output

   e. Kleene's Theorem

   f. Regular and Non-Regular Languages

   g. Context Free Grammars

   h. Pushdown Automata

   i. Context Free and Non-Context Free Languages

   j. Decidability

   k. Turing machines

XX. CSCI 412

1. CSCI 412: Operating Systems

2. 3 credits and 3 contact hours

3. Coordinator: Dr. Byunngu Yu

4. Textbook: Operating System Concepts by Abraham Silberschatz, Peter B. Galvin, Greg Gagne Publisher: Wiley; 8 edition, ISBN-10: 0470128720

    a. other supplemental materials

        i. Operating Systems: Internals and Design Principles (Fifth Edition), by William Stallings, Pub.: Prentice Hall. ISBN:0-13-147954-7

5. Specific course information

    a. Catalog description: In this course, theoretical and implementation aspects of operating system design are presented from both developer and user perspectives. Parallelism or concurrency aspects are explained using the concepts of process management, synchronization, deadlocks, job and process scheduling. Detailed techniques of real and virtual storage management are discussed for a variety of processing environments such as multiprogramming, multi-processing, etc.

    b. Prerequisites: CSCI 241/ Co-requisites: None

    c. Required course

6. Specific goals for the course

    a. Students who complete this course should be able to perform the following tasks:

        i. Understand the role and fundamental principles of operating systems

        ii. Understand theory underlying how operating systems are implemented and the implications of resulting design choices

        iii. Understand how to modify operating system code

    b. Student Outcomes

        i. SO1-B Develop methods and algorithms to solve computing problems

        ii. SO2-A Design a computing-based solution using appropriate design tools to meet a given set of requirements.

        iii. SO6-C Identify risks of computing-based solutions and describe approaches to manage them

7. Brief list of topics to be covered

   a. Introduction to operating systems

   b. Operating-System Structures.

   c. Processes.

   d. Threads.

   e. CPU Scheduling.

   f. Process Synchronization.

   g. Deadlocks.

   h. Main Memory.

   i. Virtual Memory.

   j. File-System Interface.

   k. File-System Implementation.

   l. Mass-Storage Structure.

   m. I/O Systems.

   n. Protection.

   o. Security.

   p. Distributed System Structures.

   q. Distributed File Systems.

   r. Distributed Coordination.

   s. Real-Time Systems.

   t. Multimedia Systems.

   u. The Linux System.

   v. Windows XP.

   w. Influential Operating Systems.

XXI. CSCI 415

1. CSCI 415: Computer Organization and Architecture

2. 3 credits and 3 contact hours

3. Coordinator: Dr. Byunggu Yu / Dr. Hobbs

4. Textbook: Computer Organization and Architecture, William Stallings, 10th Edition, Pearson, ISBN: 9780134101613

   a. other supplemental materials

      i. Operating Systems: Internals and Design Principles (Fifth Edition), by William Stallings, Pub.: Prentice Hall. ISBN:0-13-147954-7

5. Specific course information

   a. Catalog description: Examines arithmetic and control units, system aspects of computer memory and access control functions, input-output, and system organization.

   b. Prerequisites: CSCI 241/ Co-requisites: None

   c. Required course

6. Specific goals for the course

   a. Students who complete this course should be able to perform the following tasks:

      i. Help students gain a comprehensive view of computer design from architectural and organizational perspectives.

      ii. Understand computer arithmetic and logic and the inner-workings of the CPU

      iii. Engage students in learning the latest technologies related to parallel computing and the control unit.

   b. Student Outcomes

      i. SO1-A Apply mathematical principles (algebra, calculus and differential equation) and scientific principles to solve computing problems

      ii. SO2-B Implement a computing-based solution to a computing problem to meet a given set of requirements

      iii. SO6-A Apply computer science theory, principles and practices learned in various courses to produce a computing-based solution

7. Brief list of topics to be covered

a. Basic Concepts & Performance Issues

b. Top-level View of Computer Function and Interconnection

c. Cache Memory

d. Internal Memory Technology

e. External Memory

f. Input/Output

g. Operating System Support

h. Number System & Computer Arithmetic

i. Digital Logic

j. Instruction Set: Characteristic, Functions, Addressing modes and Formats

k. Processor Function and Structure

l. Reduced Instruction Set Computers (RISCs)

m. Parallel Processing

n. Multicore Computers

o. General Purpose Graphic Processing Units

p. Control Unit Operation & Microprogrammed Control

XXII. CSCI 434

1. CSCI 434: Analysis of Algorithms

2. 3 credits and 3 contact hours

3. Coordinator: Dr. Li Chen

4. Textbook: Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, Introduction to Algorithms, 3rd Edition (The MIT Press) 3rd Edition

   a. other supplemental materials: none

5. Specific course information

   a. Catalog description: Introduction to theoretical algorithm analysis, including study of growth rates of functions, worst-case and average behavior, and divide and conquer. Topics will include graphs, strings and dynamic programming.

   b. Prerequisites: CSCI 241, MATH 152/156 or equivalent / Co-requisites: None

   c. Required course

6. Specific goals for the course

   a. Students who complete this course should be able to perform the following tasks:

      i. Have an understanding of the various algorithmic techniques such as iteration and recursion

      ii. Have an understanding of different data structures along with their advantages/ disadvantages and resource requirements and know how to implement them.

      iii. Know how to select the most appropriate data structure and algorithm for different types of applications

      iv. Know how to analyze the performance and efficiency of algorithms

      v. Describe the classes P and NP and explain NP-completeness

   b. Student Outcomes

      i. SO1-C Estimate time and space complexity of algorithms

      ii. SO2-C Evaluate a computing-based solution by defining metrics and measuring the performance of the solution that meet a given set of requirements

      iii. SO6-C Identify risks of computing-based solutions and describe approaches to manage them

7. Brief list of topics to be covered

    a. Foundations

    b. Sorting and Order Statistics

    c. Data Structures

    d. Advanced Design and Analysis Techniques

    e. Advanced Data Structures

    f. Graph Algorithms

XXIII. CSCI 441

1. CSCI 441: Digital Forensics

2. 3 credits and 3 contact hours

3. Coordinator: Dr. Anteneh Girma

4. Textbook: Guide to Computer Forensics and Investigations, 5th Edition, Bill Nelson, Amelia Phillips, Christopher Steuart, Cengage Learning, ISBN-10: 1285060032

   a. other supplemental materials: none

5. Specific course information

   a. Catalog description: This course will teach the concepts in digital/computer forensic analysis and Internet Investigations. Specifically, this course focuses on understanding various mechanisms to detect cyber-crime, preservation of evidence, government regulations, etc. In addition, legal and technical aspects of study to achieve a balance similar to that encountered during common cases in which computer forensics are employed

   b. Prerequisites: APCT 232, 234/ Co-requisites: None

   c. Elective course

6. Specific goals for the course

   a. Students who complete this course should be able to perform the following tasks:

      i. Know the importance of digital evidence as evidence of cyber-criminality and the impact this criminality has on the global community.

      ii. Understand digital forensic techniques and be able to acquire digital evidence in a forensically sound manner

      iii. Know how to examine various logs and examine network activity

      iv. Determine where digital evidence resides on computer storage devices

      v. Know how to seize a computer from a crime scene without damaging it or risking it becoming inadmissible in a court of law.

      vi. Know how to stay current within the sub-field of digital forensics by staying aware of new developments in the field.

   b. Student Outcomes

      i. SO1-A Develop methods and algorithms to solve computing problems

    ii. SO3-A Design a computing-based solution using appropriate design tools to meet a given set of requirements.

    iii. SO4-A Apply software development fundamentals to produce a computing-based solution

7. Brief list of topics to be covered

    a. Forensics Overview

        i. Fundamentals

        ii. Technology

        iii. Vendors and Services

    b. Evidence and Capture

        i. Data Recovery

        ii. Collection and Data Seizure

        iii. Duplication and Preservation of Digital Evidence\

    c. Analysis

        i. Discovery of Electronic Evidence

        ii. Identification of Data

        iii. Reconstructing Past Events

    d. Countermeasures: Information Warfare

        i. Fighting Against Macro Threats: Defensive Strategies for Governments and Industry Groups

        ii. The Information Warfare Arsenal, Tactics

        iii. Surveillance

        iv. Civilian Casualties: The Victims and Refugees

    e. Advanced Forensics and Future Directions

    f. Case Studies from Different Domains

    g. Database Backup and Recovery

XXIV. CSCI 452

1. CSCI 452: Database System Design
2. 3 credits and 3 contact hours
3. Coordinator: Dr. Byunggu Yu
4. Textbook: Guide to Computer Forensics and Investigations, 5th Edition, Bill Nelson, Amelia Phillips, Christopher Steuart, Cengage Learning, ISBN-10: 1285060032
    a. other supplemental materials
        i. Database Fundamentals, by Byunggu Yu, Wiley (PROVIDED: Book Chapter PDF)
        ii. Database System Concept, by A. Siblerschatz, H. Korth, and S. Sudarshan, McGraw-Hill (RECOMMENDED)
        iii. Electronic Lecture Notes ("Download class notes)
5. Specific course information
    a. Catalog description: This course covers database design, entity-relationship and relational model, relational algebra, query language SQL, storage and file structures, query processing, database system architectures.
    b. Prerequisites: CSCI 241 or CSCI 343 / Co-requisites: None
    c. Required course
6. Specific goals for the course
    a. Students who complete this course should be able to perform the following tasks:
        i. Understand database system architecture, components, and services.
        ii. Understand and compare basic database design approaches
        iii. Understand database memory and storage management as a DBA
        iv. Understand user account & session management and security as a DBA
        v. Understand basic database migration, backup and recovery mechanisms as a DBA
        vi. Design, implement and manage a reliable database system, given an application
    b. Student Outcomes
        i. SO1-B Develop methods and algorithms to solve computing problems

ii. SO2-A Design a computing-based solution using appropriate design tools to meet a given set of requirements.

iii. SO3-C Produce appropriate graphics such as figures, tables in written and oral communication

iv. SO6-B Apply software development fundamentals to produce a computing-based solution

7. Brief list of topics to be covered

   a. LECTURE

      i. Database System Overview

      ii. Database Design Process

      iii. SQL

      iv. Advanced SQL

   b. LAB (No Credit Hour Self-Study)

      i. DBMS Installation

      ii. Getting Started

      iii. Managing Network Connections

      iv. Managing Memory and Storage

      v. Managing User Accounts and Sessions

      vi. Database Migration

      vii. Database Backup and Recovery

XXV. CSCI 454

1. CSCI 454: Computer Graphics

2. 3 credits and 3 contact hours

3. Coordinator: Dr. Dong H Jeong

4. Textbook: Computer Graphics with OpenGL by Hearn, Baker, and Carithers Publisher: Prentice Hall; 4th edition (Nov. 19, 2010) ISBN-10: 0136053580

   a. other supplemental materials

      i. Hearn, Donald and M. Pauline Baker, Computer Graphics, C version, 2nd Edition, Prentice Hall, 1997

      ii. Chen, Jim X., Wegman, Edward J., Foundation of 3D Graphics Programming Using JOGL and Java3D, Springer, 2006.

5. Specific course information

   a. Catalog description: This course provides an introduction to the theory and practice of computer graphics. Students are required to know basic mathematics (geometry and transformation) and basic linear algebra (such as matrix multiplication). In this course, the standard OpenGL library is going to be used to illustrate graphics theories and to show practices of computer graphics applications.

   b. Prerequisites: CSCI 241, MATH 225 / Co-requisites: None

   c. Elective course

6. Specific goals for the course

   a. Students who complete this course should be able to perform the following tasks:

      i. Understand and be able to program in a computer graphics language such as C++ with OpenGL and code programs independently and in peer groups.

      ii. Understand and be able to use algorithms for computer graphics

      iii. Understand difference between digital objects and continuous objects

      iv. Demonstrate ability in three-dimensional transformation.

   b. Student Outcomes

      i. SO1-A Apply mathematical principles (algebra, calculus, and differential equations) and scientific principles to solve computing problems

ii. SO2-B Implement a computing-based solution to a computing problem to meet a given set of requirements

iii. SO5-A Demonstrate ability to participate as a team member or leader in developing and selecting ideas, establishing team goals and objectives, and creating a collaborative and inclusive environment.

iv. SO5-B Ability to plan collaborative tasks, understand individual responsibility, share responsibilities and information on schedule, and engage in the success of team goals.

7. Brief list of topics to be covered

a. A Survey of Computer-Graphics Applications

b. Computer Graphics Hardware and Software

c. Graphics Output Primitives and Their Attributes

d. Implementation Algorithms for Graphics Primitives and Attributes

e. Two-Dimensional Viewing and Geometric Transformations

f. Three-Dimensional Viewing and Geometric Transformations

g. Hierarchical Modeling

h. Computer Animation

i. Basic Three-Dimensional Object Representations

j. Spline Representations

k. Other Three-Dimensional Object Representations

l. Visible-Surface Detection Methods

m. Illumination Models and Basic Surface-Rendering Methods

n. Paper Presentation

o. Texturing and Surface-Detail Methods

p. Color Models and Color Applications

q. Interactive Input Methods and Graphical User Interfaces

r. Global Illumination

XXVI. CSCI 455

1. CSCI 455: Cryptography
2. 3 credits and 3 contact hours
3. Coordinator: Dr. Timothy Oladunni
4. Textbook: Hacking Secret Ciphers with Python: A beginner's guide to cryptography and computer programming with Python by Al Sweigart (2013). (ISBN 13: 978-1482614374)
   a. other supplemental materials
      i. Hand-out materials will be provided.
      ii. William Stallings, Cryptography and Network Security: Principles and Practice, 6th Edition, Prentice Hall, 2013. (ISBN-10: 0133354695, ISBN-13: 978-0133354690).
5. Specific course information
   a. Catalog description: This class will provide the student a basic understanding of cryptography through algorithms. In addition, this class will cover the necessary materials including data structures, basic algorithms, computational complexity, elementary number theory, and basic cryptography including private key cryptosystems and public key cryptosystems.
   b. Prerequisites: Senior Standing / Co-requisites: None
   c. Elective course
6. Specific goals for the course
   a. Students who complete this course should be able to perform the following tasks:
      i. Understand how computational complexity relates to cryptography
      ii. Explain symmetric/asymmetric encryption and decryption
      iii. Have an ability to do basic cryptanalysis
      iv. Have knowledge on mechanics of public-key cryptography
      v. Have an ability to use public-key encryption and illustrate the difference between symmetric and public- key cryptography.
      vi. Have an ability to implement encryption and decryption algorithms
   b. Student Outcomes

i. SO1-A Apply mathematical principles (algebra, calculus, and differential equations) and scientific principles to solve computing problems

ii. SO2-C Evaluate a computing-based solution by defining metrics and measuring the performance of the solution that meet a given set of requirements

iii. SO6-C Identify risks (security, safety, market, financial, technology, people, quality, structure, and process) of computing-based solutions and describe approaches to manage them

7. Brief list of topics to be covered

    a. Review of Cryptography

    b. Classical Cryptosystems

    c. Number Theory

    d. Notation methods

    e. Modular Exponentiation

    f. Jacobi Symbols

    g. Data Encryption and DES

    h. RSA algorithm

    i. Discrete Logarithms and Diffie-Hellman Key Exchange

    j. Hash Functions

    k. Digital Signatures

    l. RAS signatures

    m. Hashing and Signing

    n. Digital signature algorithm

    o. Secure protocols

    p. X.509 certificates

    q. SSL and TLS

    r. Information Theory

    s. Entropy

t. Huffman Codes

XXVII. CSCI 498

1. CSCI 498: Senior Project I
2. 3 credits and 3 contact hours
3. Coordinator: Dr. Li Chen
4. Textbook: None
    a. other supplemental materials
        i. Various technical papers. Instructor's sample papers
5. Specific course information
    a. Catalog description: Students learn emerging topics and vocabularies in the discipline and problem-solving skills through capstone projects. This course teaches students how to continuously explore new ideas through their post-graduation life.
    b. Prerequisites: Senior Standing / Co-requisites: None
    c. Required course
6. Specific goals for the course
    a. Students who complete this course should be able to perform the following tasks:
        i. Demonstrate a knowledge of research techniques and literature survey skills by investigating the feasibility of a proposed project and its societal implications.
        ii. Know how to plan, propose, and prepare to implement a new project in the discipline.
        iii. Demonstrate communication skills and public speaking skills through written and oral presentations
        iv. Learn proposal development skills to initiate an application-oriented or research-based project in the discipline
    b. Student Outcomes
        i. SO3-A Produce a variety of written documents using appropriate formats and grammar with discipline-specific conventions including citations appropriate to the audience
        ii. SO3-C Produce appropriate graphics such as figures, tables in written and oral communications

iii. SO5-A Participate as a team member or leader in developing and selecting ideas, establishing team goals and objectives, and creating a collaborative and inclusive environment

iv. SO5-B Plan collaborative tasks, understand individual responsibility, share responsibilities and information on schedule, and engage in the success of team goals

v. SO6-A Apply computer science theory, principles and practices learned in various courses to produce a computing-based solution

vi. SO6-C Identify risks of computing-based solutions and describe approaches to manage them

7. Brief list of topics to be covered

   a. Learning new topics and vocabularies by employing proven research techniques. (Students learn new vocabularies in diverse research fields.)

   b. Understanding the current state-of-the-art in the discipline by finding relevant information, knowledge, and learning resources to make the planning project successful.

   c. Planning and proposing a new project using sound analysis and design principles in visualizing the project.

   d. Learn to find, read, and summarize. relevant technical literature.

   e. Writing the project proposal and understanding standard procedures of footnoting, referencing, and symbol usage in a technical paper.

   f. Skillful communication skills. (Enhance the ability to skillfully communicate on a technical subject to an audience less knowledgeable than the author by providing rich evidence to the senior project.)

XXVIII. CSCI 499

1. CSCI 499: Senior Project II
2. 3 credits and 3 contact hours
3. Coordinator: Dr. Li Chen
4. Textbook: None
    a. other supplemental materials
        i. Various technical papers. Instructor's sample papers
5. Specific course information
    a. Catalog description: Students learn project management skills and intensive writing skills and use the skills to professionally present the project results of Senior Project 1.
    b. Prerequisites: CSCI 498 / Co-requisites: None
    c. Required course
6. Specific goals for the course
    a. Students who complete this course should be able to perform the following tasks:
        i. Design and develop an application-oriented or research-based project of significant complexity in the discipline.
        ii. Understand the professional, ethical and social aspects of system design and creation.
        iii. Prepare a presentation, oral or written (including poster) of their project and deliver to an audience of faculty and peers.
    b. Student Outcomes
        i. SO 3-A Produce a variety of written documents using appropriate formats and grammar with discipline-specific conventions including citations appropriate to the audience.
        ii. SO3-B Deliver well-organized, logical oral presentations, including good explanations when questioned
        iii. SO3-C Produce appropriate graphics such as figures, tables in written and oral communications
        iv. SO4-A Demonstrate the knowledge of ACM Code of Ethics and Professional Conduct

v. SO5-A Participate as a team member or leader in developing and selecting ideas, establishing team goals and objectives, and creating a collaborative and inclusive environment

vi. SO5-B Plan collaborative tasks, understand individual responsibility, share responsibilities and information on schedule, and engage in the success of team goals

vii. SO6-B Apply software development fundamentals to produce a computing-based solution

7. Brief list of topics to be covered

a. Learn how to write a polished research paper by following IEEE or ACM research paper templates.

b. Understanding standard procedures of footnoting, referencing, and symbol usage

c. Writing a comprehensive project report.

d. Skillful communication (Enhance the ability to skillfully communicate on a technical subject to an audience less knowledgeable than the author by providing rich evidence to the senior project.)