Course Number: CSCI 1210

Course Name: Computer Programming Fundamentals:

Course Description:
This course is an introduction to problem-solving methods and algorithm development. Students will learn how to design, code, debug, and document programs. Students will explore basic programming concepts including variables, data types, operators and expressions. Students will learn about input/output mechanisms, including command prompt interaction, and reading and writing data to files. Students will be introduced to control structures such as branching, conditionals, iteration, and loops and arrays. They will also learn how to define and use functions/methods to structure code and improve code reuse.

Student Learning Outcomes: Students completing this course should be able to:

1. Demonstrate an understanding of procedural programming techniques by implementing programs which employ structured programming techniques.
2. Implement control flow structures in programs to execute statements in a specified order, repeat sequences of statements, and execute different statements based on conditions.
3. Apply modularization principles by defining and using functions/methods to structure code and improve code reuse and maintainability.
4. Write code utilizing data structures such as arrays, simple classes and objects, to provide useful access to, and operations on, data.
5. input/output mechanisms to collect user input and display data, including implementing error handling mechanisms to handle invalid input and output operations. the concept of recursion and identify base case and inductive step.

Credits: 4

Course Number: CSCI 1220

Course Name: Computer Programming Fundamentals: Python

Course Description:
This course is an introduction to fundamental programming concepts, with a focus on problem-solving techniques and algorithm development using the Python programming language. Students will learn how to create basic scripts, work with data types and variables, use control structures, and build functions. The course is designed for students with little or no prior experience in programming and is intended to provide a foundation in programming that can be applied in a variety of fields.

Student Learning Outcomes: Student completing this course should be able to:

1. Apply programming concepts to design and develop solutions for computational problems.
2. Utilize optimal flow-control strategy for solving a given problem.
3. Design and implement functions to support organization, decomposition and reusability of code.
4. Evaluate and select data structures to efficiently organize and store information for a given problem.
5. Demonstrate the concept of scope to control access to global, local, and class variables.
6. Create and use a simple class to demonstrate object-oriented programming principles.
7. Utilize basic file input/output with text-based files.

Credits: 3

Course Number: CSCI 1230

Course Name: **C Programming**

Course Description:
Introduction to programming in the C language. Topics include input and output, operators and expressions, control structures for branching and looping, functions, pointers and arrays. Functions in the standard libraries will be covered.

Student Learning Outcomes: Students completing this course should be able to:

1. Develop C programs that apply the functions in the C libraries to read input data and generate output.
2. Understand the various operators, and the rules of precedence, available in C, and be able to combine them to produce complex expressions to correctly calculate formulas.
3. Develop C programs that use selection statements (if-else, switch) and loop constructs (while, for do) to solve problems.
4. Modularize their C programs appropriately by using functions, with an understanding of call-by-value parameters.
5. Employ the C pre-processor features, such as #define, as appropriate.

Credits: 3

Course Number: CSCI 1240

Course Name: C++ Programming I

Course Description:
This course is an introduction to problem-solving methods and algorithm development using C++. Students will learn how to design, code, debug, and document programs. Students will explore basic programming concepts including variables, data types, operators and expressions. They will also learn how to work with the C++ preprocessor directives and libraries. Students will learn about input/output mechanisms, including command prompt interaction, reading and writing data to files. Students will be introduced to control structures such as branching, conditionals, iteration, and loops and arrays. They will also learn how to define and use functions to structure code and improve code reuse.

Student Learning Outcomes: Students completing this course should be able to:
1. Implement programs which employ structured programming techniques.
2. Implement control flow structures.
3. Apply modularization principles by defining and using functions/methods to structure code and improve code reuse and maintainability.
4. Write code utilizing data structures such as arrays, using pointers, and simple classes to provide useful access to, and operations on, data.
5. Use input/output mechanisms to collect user input and display data, including implementing error handling mechanisms to handle invalid input and output operations.

Credits: 4
Prereq: Computer Principles or a Math class.

Course Number: CSCI 1250

Course Name: Web Development

Course Description:
Introduction to HTML. Creating and designing a web page in HTML, adding tables and using frames in web design. Use XML to create interactive, dynamic Web pages. Implement cascading style sheets.

Student Learning Outcomes: Students completing this course should be able to:

1. Construct a functional webpage using HTML.
2. Plan a webpage using a site map.
3. Construct uniform style applications.
4. Develop applications in a current web programming language.
5. Demonstrate a working knowledge of the latest technologies used in creating webpages.

Credits: 3

Course Number: CSCI 1260

Course Name: Computer Programming Fundamentals Java I

Course Description: Prereq: Computer Principles or a Math class.
This course is an introduction to problem-solving methods and algorithm development using Java. Students will learn how to design, code, debug, and document programs. Students will explore basic programming concepts including variables, data types, operators and expressions. They will also learn how to work with the Java preprocessor directives and libraries. Students will learn about input/output mechanisms, including command prompt interaction, reading and writing data to files. Students will be introduced to control structures such as branching, conditionals, iteration, and loops and arrays. They will also learn how to define and use functions to structure code and improve code reuse.

Student Learning Outcomes: Students completing this course should be able to:
1. Implement programs which employ structured programming techniques.
2. Implement control flow structures.
3. Apply modularization principles by defining and using functions/methods to structure code and improve code reuse and maintainability.
4. Write code utilizing data structures such as arrays, using pointers, and simple classes to provide useful access to, and operations on, data.
5. Use input/output mechanisms to collect user input and display data, including implementing error handling mechanisms to handle invalid input and output operations.

Credit Hours: ?
Prereq: Computer Principles or a Math class.

Course Number: CSCI 2210

Course Name: Object-Oriented Programming

Course Description:
This course is an introduction to object-oriented programming. Including:
Classes and objects, and associated topics such as constructors, properties, and methods, inheritance, polymorphism, encapsulation, abstraction, exception handling and best practices.

Student Learning Outcomes: Students completing this course should be able to:

1. Implement object-oriented designs based on project requirements.
2. Use encapsulation to write programs that are loosely coupled and easy to debug, maintain, and modify.
3. Use inheritance to define simple class hierarchies that allow code to be reused by distinct subclasses.
4. Implement and reason about control flow in a program using polymorphism to solve common programming problems.

Credits: 4
Prereq: Computer Programming Fundamentals CSCI 1210.

Course Number: CSCI 2220

Course Name: Introduction to Data Structures and Algorithms

Course Description:
Design, implement, and use fundamental abstract data types including linked lists, stacks, queues, and trees. Analyze the time and space complexity of algorithms, such as sorting.

Student Learning Outcomes: Students completing this course should be able to:
1. Implement basic data structures such as linked lists, stacks, queues, and trees in a high-level programming language.
2. Compare alternative implementations of data structures with respect to time and space complexity.
3. Explain the advantages and disadvantages of a variety of sorting algorithms.

Credits: 3
Prereq: Computer Programming Fundamentals CSCI 1210.

Course Number: CSCI 2230

Course Title: Assembly Language and Machine Organization

Course Description: Computer structure and system organization, instruction execution, memory addressing modes, hardware/software interface. Programming in assembly language.

Student Learning Outcomes: Students completing this course should be able to:

1. Describe the architecture of a microcontroller, the interconnections between the components, and the major units inside the CPU.
2. Use signed and unsigned numbers, bitwise operations, branching instructions, and the corresponding flags in the status register.
3. Use immediate, direct, indirect addressing modes in assembly instructions.
4. Map high-level programming language features to assembly instructions, such as loops, conditionals, procedure calls, value and reference parameter passing, return values, and recursion.
5. Interface with input/output devices via instructions, memory addressing, or interrupts.
6. Design and implement an assembly language program.

Credits: 4 credits
Prereq: Computer Programming Fundamentals CSCI 1210.

Course Number: CSCI 2260

Course Name: Computer Programming Fundamentals: Java 2

Course Description: This course is a continuation of Java object-oriented programming. This course is a deeper exploration of object-oriented programming. Including: Classes and objects, and associated topics such as constructors, properties, and methods, inheritance, polymorphism, encapsulation, abstraction, excerption handling and best practices.

Student Learning Outcomes: Students completing this course should be able to:

1. Implement appropriate object-oriented designs based on project requirements.
2. Use encapsulation to write programs that are loosely coupled and easy to debug, maintain, and modify.
3. Use polymorphism to solve common programming problems.

Credits: ?

Prereq: Computer Programming Fundamentals: Java 1.

Course Number: CSCI 2310

Course Name: Discrete Mathematics for Computer Science

Course Description:
Discrete mathematics required for Computer Science, including topics such as the basics of logic, number theory, methos of proof, sequences, mathematical induction, set theory, counting and functions.

Student Learning Outcomes: Students completing this course should be able to:

1. Use the concept of sets and functions to reason about application domain problems.
2. Apply counting principles to determine the number of various combinatorial problems.
3. Use logic to specify precise meaning of statements, demonstrate the equivalence of statements, and test the validity of arguments.
4. Use summations, formulas for the sum of arithmetic and geometric sequences.
5. Construct and recognize valid proofs using different techniques including the principle of mathematical induction.

Credits: ?