

COMPUTER SCIENCE   
 DEPARTMENT

CS 265 - STUDENT LEARNING OUTCOMES

**Analyze the efficiency of recursive algorithms.**

* Below Expectations: With assistance interprets a recursive method.
* Meets Expectations: Analyzes a recursive method and accurately predicts its output.
* Exceeds Expectations: Evaluates recursive algorithms in terms of efficiency and time and space trade-offs.

**Assess the appropriateness of using recursion to solve a problem.**

* Below Expectations: Explains the utility of recursion to solve certain problems
* Meets Expectations: Compares and contrasts the trade-offs of recursive and non-recursive solutions.
* Exceeds Expectations: Justifies when to choose recursive solution over a non-recursive solution in terms of efficiency, Big-O, and comprehensibility.

**Compare and contrast a range of searching and sorting algorithms for time and space efficiencies.**

* Below Expectations: Uses various searching and sorting algorithms, and investigates time and space tradeoffs.
* Meets Expectations: Compares and contrasts a range of searching and sorting algorithms for time and space efficiencies.
* Exceeds Expectations: Critiques searching and sorting algorithms, including recursive solutions, for various algorithmic efficiencies and evaluates them in terms of Big-O.

**Create effective, efficient, and secure software reflecting standard principles of software assurance and software engineering.**

* Below Expectations: Calculates the risks and liabilities of a computer-based solution using standard software assurance and engineering principles.
* Meets Expectations: Creates an effective, efficient and secure solution, utilizing principles of software assurance and software engineering.
* Exceeds Expectations: Judges the safety and security of a software solution.

**Discuss and construct programming solutions using a variety of recursive techniques.**

* Below Expectations: Converts a simple recursive algorithm into a working recursive method.
* Meets Expectations: With guidance develops recursive programming solutions for applications that use data structures such as trees and lists.
* Exceeds Expectations: Independently designs and develops recursive programming solutions for applications that use backtracking and data structures such as trees and lists.

**Design and develop reusable software using appropriate data structures and templates.**

* Below Expectations: Differentiates among the classic data structures and selects a suitable data structure for use in an application.
* Meets Expectations: With some guidance designs and develops applications using appropriate data structures for a given problem.
* Exceeds Expectations: Independently designs and develops applications using appropriate data structures and incorporates reusable software components in the solution.

**Proactive the tenets of ethics and professional behavior promoted by computing societies; accept the professional responsibilities and liabilities associated with software development.**

* Below Expectations: Studies the tenets of ethics and professional behavior promoted by international computing societies, such as ACM and IEEE-CS.
* Meets Expectations: Practices the tenet of ethics and professional behavior promoted by international computing societies, and recognizes the liabilities associated with software development.
* Exceeds Expectations: Displays ethical and professional behavior associated with the responsibilities of software development.

**Use standard analysis and design techniques to produce a team developed, medium sized, secure software application that is fully implemented and formally tested.**

* Below Expectations: As part of a team, produces an executable, medium-sized software application that meets some program requirements and includes design documentation and some evidence of testing.
* Meets: As part of a team, produces a working, medium-sized software application on time that meets many program requirements including design and some test plan documentation.
* Exceeds Expectations: As part of a team, successfully develops a medium-sized, secure software application on time that meets all program requirements including design and formal test plan documentation.