Syllabus – Fall term, 2022-23

Student outcomes (learning objectives):

Students who successfully complete this course should be able to:

Analyze, explain and use appropriately in coding: ***Fundamental programming concepts*** **including:**

1. Syntax and semantics
2. Objects, types, names (variables), expressions, and assignment
3. Branching control structures
4. Explicit loops, both definite and indefinite
5. Functions, parameter passing, user-defined functions
6. Constructing objects, and using their methods and instance variables (fields)
7. Components of a class, as expressed in code as well as in Unified Modeling Language (UML) or other such diagrams
8. Sequences, including lists and strings
9. Indirection, box-and-pointer diagrams and mutable objects
10. Input and output, to both consoles and text files
11. Modularity and structured decomposition to break a program into smaller pieces
12. Using an application programming interface (API)

Design, implement, debug and test small programs for solving problems motivated by real-world interests, using the *above concepts* and ***modern software engineering practice****s* including (where appropriate, and at an elementary level):

1. An appropriate integrated development environment with version control
2. Coding to a specification
3. Iterative enhancement
4. Pair programming
5. Test-first programming
6. Documenting software, for internal readers and for external readers
7. Use of application programming interfaces (APIs)

Work for 2 - 4 weeks in a team of 3-4 students on a small software development project, demonstrating (at an elementary level) effective use of:

1. Division of labor
2. Integrating teammates' work
3. Modularity and structured decomposition to break a program into smaller pieces
4. Constructing objects from new APIs as needed, and using their methods and instance variables (fields)
5. Agile software development processes
6. Team roles
7. Conflict resolution