**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:

## Syntax and semantics

## Objects, types, names (variables), expressions, and assignment

## Branching control structures

## Explicit loops, both definite and indefinite

## Functions, parameter passing, user-defined functions

## Constructing objects, and using their methods and instance variables (fields)

## Components of a class, as expressed in code as well as in Unified Modeling Language (UML) or other such diagrams

## Sequences, including lists and strings

## Indirection, box-and-pointer diagrams and mutable objects

## Input and output, to both consoles and text files

## Modularity and structured decomposition to break a program into smaller pieces

## 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 practices*including (where appropriate, and at an elementary level):

## An appropriate integrated development environment with version control

## Coding to a specification

## Iterative enhancement

## Pair programming

## Test-first programming

## Documenting software, for internal readers and for external readers

## 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:

## Division of labor

## Integrating teammates' work

## Modularity and structured decomposition to break a program into smaller pieces

## Constructing objects from new APIs as needed, and using their methods and instance variables (fields)

## Agile software development processes

## Team roles

## Conflict resolution