1. **Course number and name**
   CptS 321: Object-Oriented Software Principles

2. **Credits and contact hours**
   3 credits, 3 lecture hours

3. **Instructor’s or course coordinator’s name**
   Venera Arnaoudova

4. **Textbook, title, author, and year**

5. **Specific course information**
   a. **Catalog description**: Object-oriented programming for flexibility, efficiency, and maintainability; logic and UI decoupling; complexity analysis, data structures, and algorithms for industry-quality software.
   b. **Prerequisites or corequisites**: Cpt S 223 with a C or better or Cpt S 233 with a C or better; certified major in Computer Science, Computer Engineering, Electrical Engineering, or Software Engineering.

6. **Specific goals for the course**
   By the end of the course, students will be able to
   - Describe and apply the fundamental concepts of the C# programming language including the typing system, inheritance and polymorphism rules, and reflection (1a, 1c, 1e, 2a, 2b, 2c, 2d, 2e, 2g).
   - Describe how the event-driven model works for desktop applications and apply this knowledge to build software with graphical user interfaces (1a, 1c, 1e, 2a, 2b, 2c, 2d, 2e, 2g).
   - Describe and implement object-oriented design patterns and strategies that allow for updates to code that do not disrupt any other existing application components (1a, 1c, 1e, 2a, 2b, 2c, 2d, 2e, 2g).
   - Explain and implement expression tree data structures for evaluating arithmetic expressions (1a, 1c, 1e, 2a, 2b, 2c, 2d, 2e, 2g).
   - Build applications piece-by-piece, building on previous foundations without making the code difficult to manage (1a, 1c, 1e, 2a, 2b, 2c, 2d, 2e, 2g).

7. **Brief list of topics to be covered**
   - Decoupling of software components.
   - Data structures for modern software features.
   - Design principles.
   - Introduction to Graphical User Interface (GUI) design.
   - Event-driven applications.
   - C# language features, streams.
   - Arithmetic expression trees.
- Spreadsheet application (formula evaluation, loading/saving).
- Design choices in .NET framework objects.
- Undo/redo system design.
- C# extension methods.
- XML.
- Threading.
- WinForms custom controls.