1. **Course number and name**
   CptS 476: Software Construction and Maintenance

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

3. **Instructor's or course coordinator's name**
   Sakire Arslan Ay

4. **Textbook, title, author, and year**
   I. Sommerville. 2011. *Software Engineering* (9th ed.). Addison-Wesley.
   S. McConnell, 2004. *Code Complete: A Practical Handbook of Software Construction* (2nd ed.). Microsoft Press. (Recommended)
   P. Grubb and A.A. Takang, 2003. *Software Maintenance: Concepts and Practice* (2nd ed.). World Scientific Publishing. (Recommended)
   IEEE Std. 14764-2006 (a.k.a. ISO/IEC 14764:2006) Standard for Software Engineering—Software Life Cycle Processes—Maintenance, IEEE, 2006.

5. **Specific course information**
   a. *Catalog description: "*Software quality, construction (API design and use, object-oriented runtime issues), and maintenance (refactoring, reengineering, reverse engineering)."
   b. *Prerequisites or corequisites:* CPTS 322 with a C or better; certified major in CptS, CptEE, EE, SE, or DA.

6. **Specific goals for the course**
   By the end of the course, students will be able to:
   - Explain purposes of metrics, quality processes, methods for measuring that quality, and standards used.
   - Work with a team to build a software system and add/modify features of an existing software system (2c, 2d, 2g, 5b, 5d, 5g).
   - Apply common construction and maintenance heuristics to enhance existing code, such as ways to eliminate global variables and ways to test difficult code (6a, 6c).
   - Organize and develop software user documentation which enhances long-term software viability (3a, 3b, 3c).
   - Construct software so that it meets delivery and deployment objectives specified by the project (6a, 6b).
   - Apply the corrective, perfective, adaptive and preventive types of software changes and maintenance types (1e).
   - Apply impact analysis and other software source analysis to understanding existing software (4a).
   - Describe software modernization approaches such as reverse engineering, reengineering, and restructuring.

7. **Brief list of topics to be covered**

- Introduction to Software Quality.
- Value and cost of quality.
- Models and quality characteristics.
- Quality measurement and improvement.
- Software construction fundamentals (e.g., minimizing complexity, anticipating changes, reuse).
- Construction design, languages, coding (e.g., code conventions, documentation).
- Construction testing, reuse, quality, and integration.
- Construction technologies (e.g., design and use, generics, assertions, design by contract, defensive programming, error/exception handling).
- Construction tools (e.g., integrated development environment (IDE), graphical user interface (GUI) builders, testing, profiling, performance analysis).
- Introduction to software maintenance and evolution. Fundamentals (definition, nature, categories of maintenance).
- Key issues during maintenance (technical, management, cost).
- Maintenance process.
- Program comprehension.
- Reengineering.
- Reverse engineering.