Course Title: Software Testing Fundamentals

Course Description: This course provides an in-depth introduction to software testing, covering fundamental principles, techniques, and best practices. Students will gain hands-on experience in planning, designing, and executing test cases, as well as reporting and managing defects. By the end of the course, students will be equipped with the skills needed to ensure the quality and reliability of software applications.

Course Duration: 12 weeks (4 hours per week)

Course Objectives: By the end of this course, students should be able to:

1. Understand the importance of software testing in the software development lifecycle.
2. Identify and analyze different software testing levels and types.
3. Develop effective test plans and strategies.
4. Create test cases and test scripts.
5. Execute tests manually and using automation tools.
6. Report and manage defects using a bug tracking system.
7. Apply best practices in software testing.
8. Collaborate effectively with development and QA teams.
9. Gain practical experience in testing real-world applications.

Course Outline:

Week 1-2: Introduction to Software Testing

- What is software testing?
- Importance of software testing
- Testing principles and objectives

- Testing terminology and concepts

Week 3-4: Testing Levels and Types

- Unit testing
- Integration testing
- System testing
- Acceptance testing
- Functional vs. non-functional testing
- Regression testing
- Smoke and sanity testing

Week 5-6: Test Planning and Strategy

- Test planning process
- Test strategy and approach
- Test estimation and budgeting
- Risk-based testing
- Test documentation

Week 7-8: Test Design

- Test case design techniques (boundary value analysis, equivalence partitioning, etc.)
- Test data preparation
- Traceability matrix
- Test environment setup

Week 9-10: Test Execution

- Manual testing vs. automation testing

- Test execution process
- Defect reporting and management
- Test reporting and metrics
- Continuous integration and testing

Week 11-12: Best Practices and Advanced Topics

- Test automation tools and frameworks
- Performance testing
- Security testing
- Usability testing
- Exploratory testing
- Agile and DevOps testing

Assessment:

- Weekly quizzes and assignments
- Mid-term and final exams
- Practical lab exercises
- Group projects

Note: This is a general outline, and the course content can be adjusted based on the level of the students, the available resources, and the specific goals of the course. Additionally, incorporating practical exercises, case studies, and guest speakers from industry experts can enhance the learning experience.