Pokhara University
Faculty of Science and Technology

Course Code.: CMP 440 (3 Credits)	Full marks: 100
Course Title: Software Testing, Verification Validation and Quality Assurance **(3-1-2)**	Pass marks: 45
Nature of the course: Theory & Practice	Time per period: 1 hour
Semester: VII	Total periods: 45
Level: Bachelor	Program: BE Software

1. Course Description

This course explores the goals of testing, quality assurance, and quality control activities performed during the lifecycle of a software product. The course module focuses on integrating test processes with established software development methodologies. It further emphasizes practical exercises to provide students with experience in the design, specification, and execution of tests plus test automation using tools through a combination of instructor-directed exercises and student research leading to knowledge sharing.

2. General Objectives

The general objectives of this course are

- To familiarize students with software testing as a fundamental software development life cycle component.
- To enhance the student's understanding of various types of software testing.
- To equip students with knowledge and skills to efficiently perform Testing and Quality Assurance activities using modern software tools.
- To acquaint the students with various automated testing tools.
- To provide the students with relevant knowledge of the scope of software projects.
- To expose students to various levels of software testing and quality assurance.

3. Contents in Detail

Specific Objectives	Contents
● Basic introduction to Software Testing, QA, and QC ● Introduction to different types of testing	**Unit I: Title (4 Hrs)** 1.1 Background & Importance of Testing 1.2 Introduction to Quality Control and Quality Assurance 1.3 Manual and Automated Testing 1.4 Difference between Product and Project 1.5 Software Development Lifecycle (SDLC) 1.6 Roles and Responsibilities of Software Tester, Quality Controller, Quality Analysts, DevOp Team, and, Project Manager
● SDLC stages and their significance ● In-depth study of different types of software testing ● Identifying Software Bugs and Defects	**Unit II: SDLC & Testing (8 Hrs)** 2.1 SDLC stages 2.2 Software development methodologies 2.3 Types of Testing 2.3.1 Functional or Dynamic Testing 2.3.2 Non-functional Testing 2.4 Levels of Testing 2.4.1 Unit Testing 2.4.2 Integration Testing 2.4.3 System Testing 2.4.4 Acceptance Testing 2.5 Risk of inadequate testing 2.6 Test Platforms 2.6.1 Development 2.6.2 Quality Control 2.6.3 UAT 2.6.4 Production stage 2.7 Software Bug 2.7.1 Difference between Bug and Defect 2.7.2 Bug Lifecycle 2.7.3 Defect Management Process
● Understanding the need and importance of various documents ● Various stages in Software Testing Process	**Unit III: Requirement Document (6Hrs)** 3.1 Importance of requirement documents 3.1.1 Software Requirement Specification 3.1.2 Business Requirement Document 3.1.3 Function Requirement Document 3.2 Software Testing Life Cycle (STLC) 3.3 System User Manual Doc and Preparation 3.4 Software Release Note and Preparation 3.5 Difference between Development, Test, Staging, and, Production Environment

● Performance Testing and its various types ● Significance of Testing Data ● Management of Testing Data	**Unit IV: Types of Testing & Test Cases (8 Hrs)** 4.1 Dynamic Testing and Functional Testing 4.2 Performance Testing and Its Importance 4.2.1 Load Testing 4.2.2 Stress Testing 4.2.3 Spike Testing 4.2.4 Scalability Testing 4.2.5 Volume Testing 4.3 Performance Testing Process and Example Test Cases 4.4 Regression Testing, Sanity Testing, Smoke Testing, Compatibility Testing, and Data Flow/Control Testing 4.5 Design Testing and Multi-Domain Testing 4.6 Managing Test Data and its advantages 4.6.1 Gathering Test Data 4.6.2 Creating Data Repositories
● To understand the prerequisites of software testing ● Analyse Software Test Result	**Unit V: Quality Assurance Phases & Test Plans(6 Hrs)** 5.1 Feature Requirement, Test Plan, Test Scenario, Test Cases, Test Data, Test Script, and, Test Results 5.2 Test Plan and its content 5.3 Entry and Exit Criteria 5.4 Test Coverage 5.5 Ad-Hoc Testing and Exploratory Testing
● Understand the need of an Automation Testing Tool ● Basic introduction to Selenium	**Unit VI: Manual & Automated Testing (8 Hrs)** 6.1 Automated vs Manual Testing 6.2 Benefits of Automated Testing 6.3 Automated Testing Tools and Tool Selection Criteria 6.4 Creating and Executing basic tests, recording, understanding, and executing a test 6.5 Introduction to Selenium 6.5.1 Fundamentals of Selenium 6.5.2 Export Features 6.5.3 Selenium Webdriver 6.5.4 Selenium IDE
● API Testing and Mobile Testing Introduction ● Learning LoadRunner and Postman	**Unit VII: API Testing & Mobile Testing (5 Hrs)** 7.1 API Testing through Endpoints and Its Importance 7.2 Various Methods and Types of Error Codes 7.3 Automation using LoadRunner, CSV or JSON File & Command Line

	7.4 Introduction to Postman Testing Platform 7.5 API Automation Test Report Generation 7.6 Mobile Testing & Mobile Testing Strategy 7.7 Simulators and Emulators

Note: The figures in the parentheses indicate the approximate periods for the respective units.

4. List of Tutorials

The following tutorial activities of 15 hours per group of a maximum 24 students should be conducted to cover all the required contents of this course.

S.N.	Tutorials
1	Analyze various types of software testing and determine their importance
2	Create a test case and specify what type of testing should be done.
3	Create various types of requirement documents.
4	Perform load testing of a web application, produce test results and explain the test results.
5	Create entry and exit criteria for various test cases.
6	Analyze the need for automation testing in software testing. Also, study various automation tools available.
7	Create a dummy API endpoints and create a strategy to perform API testing.

5. Practical Works

S.N.	Practical works
1	Create test scenarios and test cases.
2	Using JMeter, test various performance indexes of an application and produce the test report.
3	Create a simple test case and write a test script in any preferred language. Prepare a report showing the total test result and test coverage.
4	Use Selenium IDE to record browser activities and create test cases.
5	Using any programming language write a test case using Selenium Webdriver to automate the browser and perform basic tests.
6	Using CLI generate test reports.
7	Use Postman to create collections and environment variables. Learn to share the collections and collaborate with other testers. Write test snippets and produce test results.
9	Use simulators and emulators to test a dummy mobile application.

6. **Evaluation System and Students' Responsibilities**

Evaluation System

In addition to the formal exam(s) conducted by the Office of the Controller of Examination of Pokhara University, the internal evaluation of a student may consist of class attendance, class participation, quizzes, assignments, presentations, written exams, etc. The tabular presentation of the evaluation system is as follows.

External Evaluation	**Marks**	**Internal Evaluation**	**Marks**
Semester-End Examination	50	Class attendance and participation	5
		Lab works and reports	5+5
		Quizzes/assignments and presentations	10
		Internal Term Exam	25
Total External	50	Total Internal	50
Full Marks 50 + 50 = 100			

Students' Responsibilities:
Each student must secure at least 45% marks in the internal evaluation with 80% attendance in the class to appear in the semester-end examination. Failing to obtain such a score will be given NOT QUALIFIED (NQ) and the student will not be eligible to appear in the End-Term examinations. Students are advised to attend all the classes and complete all the assignments within the specified period. If a student does not attend the class(es), it is his/her sole responsibility to cover the topic(s) taught during the period. If a student fails to attend a formal exam, quiz, test, etc. there won't be any provision for a re-exam.

7. Prescribed Books and References

Text Book

Daniel Galin, *Software Quality Assurance: From Theory to Implementation*, Addison Wesley, 2004

Stephen Kan, *Metrics and Models in Software Quality Engineering (2nd Edition)*, Addison Wesley, 2002

Reference Book

Schulmeyer, G. Gordon and McManus, James, (eds), *Handbook of Software Quality Assurance*, 3rd Pretence Hill