



# IT3205: Fundamentals of Software Engineering (Compulsory)

**BIT – 2<sup>nd</sup> Year  
Semester 3**



## Learning Outcome

After successfully completing this course students should be able to;

- explain the software engineering principles and techniques that are used in developing quality Software products
- apply software engineering principles and techniques appropriately to develop a moderately complex software system

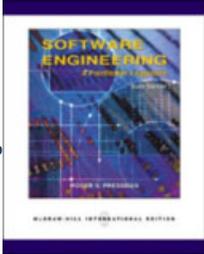
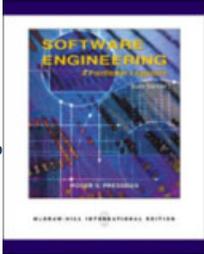


# Outline of Syllabus

1. Introduction
2. Software Development Process Models
3. Requirements Analysis & Specification
4. Design
5. Coding
6. Software Testing and Quality Assurance
7. Software Maintenance
8. Software Project Management

# Main References

- 1. Software Engineering by Ian Sommerville, 7<sup>th</sup> edition, Addison-Wesley, 2006.

The cover of the book "Software Engineering" by Ian Sommerville, 7th edition, Addison-Wesley, 2006. The cover is primarily blue with a yellow top section and an orange bottom section. The title "Software Engineering" is written in white on the blue background, with "Seventh Edition" and the author's name "Ian Sommerville" below it.
- 2. Software Engineering: A practitioner's approach by Roger S. Pressman, 6<sup>th</sup> edition, McGraw-Hill International edition, 2005.

The cover of the book "Software Engineering: A Practitioner's Approach" by Roger S. Pressman, 6th edition, McGraw-Hill International Edition, 2005. The cover features a blue background with a grid pattern and a central image of a computer monitor displaying a software interface. The title "SOFTWARE ENGINEERING" is written in yellow at the top, and "A Practitioner's Approach" is written in white below it. The author's name "ROGER S. PRESSMAN" is at the bottom.
3. <http://www.softwareengineering-9.com>



# IT3205: Fundamentals of Software Engineering

## 1. Introduction

Duration: 4 hours



## Learning Objectives

- Describe what software is, different types of software and software quality attributes
- Describe with the problems associated with software and software development
- Define what software engineering is and explain why it is important
- State some professional issues related to software development



# Detailed Syllabus

## 1.1 Software

- 1.1.1 What is software?
- 1.1.2 Types of software
- 1.1.3 Characteristics of Software
- 1.1.4 Attributes of good software

## 1.2 Software Engineering

- 1.2.1 What is software engineering?
- 1.2.2 Software engineering costs
- 1.2.3 What are the key challenges facing software engineering?
- 1.2.4 Systems engineering & software Engineering
- 1.2.5 Professional Practice



# IT3205: Fundamentals of Software Engineering

## 2. Software Development Process Models

Duration: 8 hours



## Learning Objectives

- Describe different process models used for software development
- Identify the most appropriate software process model for a given problem
- Identify how CASE tools can be used to support software process activities



# Detailed Syllabus

2.1 What is a software process?

2.2 What is a software process model?

2.2.1 The waterfall model

2.2.2 Evolutionary development

2.2.3 Component-Based Software Engineering (CBSE)

2.3 Process Iteration

2.3.1 Incremental delivery

2.3.2 Spiral development



# Detailed Syllabus

## 2.4 Rapid software development

- 2.4.1 Agile methods
- 2.4.2 Extreme programming
- 2.4.3 Rapid application development (RAD)
- 2.4.4 Software prototyping

## 2.5 Rational Unified Process (RUP)

## 2.6 Computer Aided Software Engineering (CASE)

- 2.6.1 Overview of CASE approach
- 2.6.2 Classification of CASE tools



# IT3205: Fundamentals of Software Engineering

## 3. Requirements Analysis & Specification

Duration: 8 hours



## Learning Objectives

- Identify the types of requirements, which should be captured in a software project.
- Describe and apply different requirement analysis and specification techniques.
- Prepare a software requirement specification (SRS) for a given software problem.



# Detailed Syllabus

## 3.1 System and software requirements

## 3.2 Types of software requirements

3.2.1 Functional and non-functional requirements

3.2.2 Domain requirements

3.2.3 User requirements



# Detailed Syllabus

## 3.3 Elicitation and analysis of requirements

3.3.1 Overview of techniques

3.3.2 Viewpoints

3.3.3 Interviewing

3.3.4 Scenarios

3.3.5 Use-cases

3.3.6 Ethnography

## 3.4 Requirements validation

## 3.5 Requirements specification



# IT3205: Fundamentals of Software Engineering

## 4. Software Design

Duration: 8 hours



## Learning Objectives

- Describe the important software design issues and concepts.
- Compare different approaches to software design.
- Identify suitable design approaches for a problem.



# Detailed Syllabus

## 4.1 Design concepts

4.1.1 Abstraction

4.1.2 Architecture

4.1.3 Patterns

4.1.4 Modularity

4.1.4.1 Cohesion

4.1.4.2 Coupling

4.1.5 Information hiding

4.1.6 Functional independence

4.1.7 Refinement



# Detailed Syllabus

## 4.2 Architectural design

- 4.2.1 Repository model
- 4.2.2 Client-server model
- 4.2.3 Layered model
- 4.2.4 Modular decomposition

## 4.3 Procedural design using structured methods



# Detailed Syllabus

## 4.4 User Interface design

- 4.4.1. Human-computer interaction
- 4.4.2. Information presentation
- 4.4.3. Interface evaluation

## 4.5 Design notations



# IT3205: Fundamentals of Software Engineering

## 5. Coding

Duration: 3 hours



## Learning Objectives

- Select appropriate programming language and development tools for a given problem
- Identify the features of a good program, good programming practices and program documentation



# Detailed Syllabus

- 5.1 Programming languages and development tools
- 5.2 Selecting languages and tools
- 5.3 Good programming practices



# IT3205: Fundamentals of Software Engineering

## 6. Software Testing and Quality Assurance

Duration: 8 hours



# Learning Objectives

- State the software testing process and required documentation
- Explain the different software testing techniques and integration strategies
- Design test cases and write test programs for a given simple software problem
- Describe the code verification techniques
- Describe the importance of software quality
- Distinguish the difference between product quality and process quality
- Describe some important quality standards with respect to software



# Detailed Syllabus

6.1 Verification and Validation

6.2 Techniques of testing

6.2.1 Black-box and White-box testing

6.2.2 Inspections

6.3 Levels of testing

6.3.1 Unit testing

6.3.2 Integration Testing

6.3.3 Interface testing

6.3.4 System testing

6.3.5 Alpha and beta testing



# Detailed Syllabus

## 6.3 Levels of testing

6.3.6 Regression testing

6.3.7 Back-to-back testing and Thread testing

6.3.8 Statistical Software Testing

6.3.9 Object Oriented Testing

## 6.4 Design of test cases

## 6.5 Quality management activities

## 6.6 Product and process quality

## 6.7 Standards

6.7.1 ISO9000

6.7.2 Capability Maturity Model (CMM)



# IT3205: Fundamentals of Software Engineering

## 7. Software Maintenance

Duration: 3 hours



## Learning Objectives

- Describe the types of software maintenance
- Describe the software maintenance process
- Describe activities of configuration management



# Detailed Syllabus

## 7.1 Evolving nature of software

### 7.1.1 Different types of maintenance

7.1.1.1 Fault repair

7.1.1.2 Software adaptation

7.1.1.3 Functionality addition or modification

### 7.1.2 Maintenance prediction

### 7.1.3 Re-engineering



# Detailed Syllabus

## 7.2 Configuration Management (CM)

- 7.2.1 Importance of CM
- 7.2.2 Configuration items
- 7.2.3 Versioning



# IT3205: Fundamentals of Software Engineering

## 8. Software Project Management

Duration: 3 hours



## Learning Objectives

- State the requirement of managerial control of the development process.
- Describe the main phases of software project management.
- Describe project planning and project scheduling activities in detail.



# Detailed Syllabus

- 8.1 Need for the proper management of software projects
- 8.2 Management activities
  - 8.2.1 Project planning
  - 8.2.2 Estimating costs
  - 8.2.3 Project scheduling
  - 8.2.4 Risk management
  - 8.2.5 Managing people



# FINAL EXAMINATION

- Two hours question paper
  - 25 MCQs (one hour)



**UNIVERSITY OF COLOMBO, SRI LANKA**

**UNIVERSITY OF COLOMBO SCHOOL OF COMPUTING**

**DEGREE OF BACHELOR OF INFORMATION TECHNOLOGY (EXTERNAL)**  
*Academic Year 2012/2013 – 2nd Year Examination – Semester 3*

***IT3204: Software Engineering I***

***PART I – Multiple Choice Question Paper***

***02<sup>nd</sup> March 2013***

***(ONE HOUR)***

**Important Instructions:**

- The duration of the paper is **1 (one) hour**.
- The medium of instruction and questions is English.
- The paper has **25** questions and **6** pages.
- All questions are of the MCQ (Multiple Choice Questions) type.
- All questions should be answered.
- Each question will have 5 (five) choices with **one or more** correct answers.
- All questions will carry equal marks.
- There will be a penalty for incorrect responses to discourage guessing.
- The mark given for a question will vary from 0 (*All the incorrect choices are marked & no correct choices are marked*) to +1 (*All the correct choices are marked & no incorrect choices are marked*).
- Answers should be marked on the special answer sheet provided.
- Note that questions appear on both sides of the paper.  
If a page is not printed, please inform the supervisor immediately.

Mark the correct choices on the question paper first and then transfer them to the given answer sheet which will be machine marked. **Please completely read and follow the instructions given on the other side of the answer sheet before you shade your correct choices.**



# FINAL EXAMINATION

- Two hour question paper
  - 25 MCQs (one hour)
  - 2 Mandatory structured questions (one hour)



UNIVERSITY OF COLOMBO, SRI LANKA

UNIVERSITY OF COLOMBO SCHOOL OF COMPUTING

DEGREE OF BACHELOR OF INFORMATION TECHNOLOGY (EXTERNAL)

Academic Year 2012/2013 – 2<sup>nd</sup> Year Examination – Semester 3

**IT3204: Software Engineering I  
PART 2 - Structured Question Paper**

02<sup>nd</sup> March, 2013  
(ONE HOUR)

To be completed by the candidate

BIT Examination Index No: .....

**Important Instructions:**

- The duration of the paper is **1 (one) hour**.
- The medium of instruction and questions is English.
- This paper has **2 questions** and **8 pages**.
- **Answer both questions.**
- **Both questions carry equal marks.**
- **Write your answers** in English using the space provided **in this question paper**.
- Do not tear off any part of this answer book.
- Under no circumstances may this book, used or unused, be removed from the Examination Hall by a candidate.
- Note that questions appear on both sides of the paper.  
If a page is not printed, please inform the supervisor immediately.

**Questions Answered**

Indicate by a cross (x), (e.g. ) the numbers of the questions answered.

To be completed by the candidate by marking a cross (x).	Question numbers	
	1	2
To be completed by the examiners:		





# Answer sheet for Paper I



## How to Answer?

- Read the question very carefully.
  - Remember, every word in the question have a specific meaning.
- Understand the question correctly.
  - Pay attention on the wordings of the question
- Write the answer ONLY for the question.
  - Do not waste your time on writing the background



## How to answer?

- Distinguish the meaning of following terms;
  - Define
  - Briefly define
  - Discuss
  - Briefly explain
  - Describe
  - Briefly describe



## How to answer? - Examples

- Explain some of the limitations of software testing.
  - what is testing, Alpha testing, beta testing, black-box testing, testing methods ???
  - Perfect answer is something like;  
It is very difficult to trace out logical errors through Testing.  
Stress testing or load tests are not the realistic options and hence, it cannot be defined that how application or module will be reacting at heavy data loads.  
In Integration testing, skeletons of different modules are used, which cannot describe the full functioning and in-turn the complete behavior of module they are representing.  
Being performed at later stages, testing may lead to a complete redevelopment of the module under testing and hence putting all effects in vain.



## How to answer? - Examples

- Briefly describe 3 types of software maintenance.  
what is maintenance, why maintenance is important, how to do maintenance ???

Perfect answer would be a short explanation about

Corrective maintenance

Adaptive maintenance and

Perfective maintenance



# Clue for the Exam!!!



# Clue for the Exam!!!

- Understand the subject well!



# Clue for the Exam!!!

- Understand the subject well!
- Do past papers



## Clue for the Exam!!!

- Understand the subject well!
- Follow the practice quizzes on each section
- Check your answers with given ones and try to understand what is wrong with your answer!



**GOOD LUCK!!!**