**BANGLADESH TECHNICAL EDUCATION BOARD**

Agargaon, Dhaka-1207

4-YEAR DIPLOMA-IN-ENGINEERING PROGRAM

SYLLABUS (PROBIDHAN-2016)

# COMPUTER TECHNOLOGY

TECHNOLOGY CODE: **666**

6th SEMESTER

DIPLOMA IN ENGINEERING

PROBIDHAN-2016

# COMPUTER TECHNOLOGY

## 6th Semester

| Sl. No. | Subject Code | Name of the Subject | T | P | C | Marks | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | Theory | | Practical | | Total |
| | | | | | | Cont. Assess | Final Exam | Cont. Assess | Final Exam | |
| 1 | 66661 | Principals of Software Engineering | 2 | 6 | 4 | 40 | 60 | 50 | 50 | 200 |
| 2 | 66662 | Microprocessor & Interfacing | 2 | 3 | 3 | 40 | 60 | 25 | 25 | 150 |
| 3 | 66663 | Microcontroller Application | 0 | 6 | 2 | - | - | 50 | 50 | 100 |
| 4 | 66664 | Database Management System | 2 | 3 | 3 | 40 | 60 | 25 | 25 | 150 |
| *5 | 6666X | Optional Subject -1 | 2 | 3 | 3 | 40 | 60 | 25 | 25 | 150 |
| 6 | 69054 | Environmental Studies | 2 | 0 | 2 | 40 | 60 | - | - | 100 |
| 7 | 65852 | Industrial Management | 2 | 0 | 2 | 40 | 60 | - | - | 100 |
| | | **Total** | **12** | **21** | **19** | **240** | **360** | **175** | **175** | **950** |

*** 6666X Optional Subjects-I**

| Group | Subject code | Subject Name |
|---|---|---|
| Network Maintenance Group | 66665 | Network & Data Center Operation |
| Automation System Group | 66666 | PLC Automation System |
| Software Developer Group | 66667 | Web Mastering |
| Multimedia Developer Group | 66668 | Multimedia & Animation |

## 66661 Principles of Software Engineering

| T | P | C |
|---|---|---|
| 2 | 6 | 4 |

## OBJECTIVES

• To study the approaches of application of engineering to software.
• To develop knowledge and skill to apply systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software.

## SHORT DESCRIPTION

Concept of software engineering, Basics of Software development life cycle (SDLC), Project management, Requirements analysis, Design basics, Analysis & Design tools, Design strategies, User Interface design, understanding of Design complexity, Software implementation, Testing and quality assurance, Maintenance, CASE tools overview;

## DETAIL DESCRIPTION

### Theory:

**1. Understand the concept of software engineering**

1.1 Define software engineering.
1.2 Describe the evolution of software engineering.
1.3 List software evolution laws.
1.4 Describe E-Type software evolution laws.
1.5 Describe software paradigms.
1.6 Necessity of software engineering.
1.7 List the characteristics of good software.

**2. Understand the basics of software development life cycle (SDLC)**

2.1 Describe the software development life cycle activities.
2.2 Describe software development paradigm (Waterfall model, Iterative model, spiral model, agile development)
2.3 Describe agile development.
2.4 State the agile manifesto.
2.5 List agile manifesto items.
2.6 List key principles of agile.
2.7 Describe agile methodologies

**3. Understand the software project management**

3.1 State the need of software project management.
3.2 Describe role of software project manager.
3.3 List software management activities.
3.4 Describe configuration management.
3.5 Describe project management tools.

**4. Understand software requirement engineering**

4.1 Describe software requirement engineering process.
4.2 List requirement elicitation process.
4.3 Describe requirement elicitation techniques.
4.4 List software requirements characteristics.
4.5 Describe types of software requirements.
4.6 Describe the role of software system analyst.
4.7 List software metrics and measures.

**5. Understand the software design basics, analysis and design tools**

5.1 Describe software design levels.
5.2 State modularization and concurrency.
5.3 State coupling and cohesion
5.4 Describe design verification.
5.5 State data flow diagram, structure charts.
5.6 Describe Hierarchical Input Process Output (HIPO) diagram.
5.7 State pseudo code.
5.8 Describe decision table.
5.9 Describe entity relationship model.
5.10 State data dictionary.

**6. Understand software design strategies**

6.1 Define structured design.
6.2 Describe function-oriented design.
6.3 Describe object oriented design.
6.4 Describe software design patterns.
6.5 Describe software design approaches.

**7. Understand user interface design**

7.1 Describe command line interface (CLI).
7.2 Describe graphical user interface (GUI).
7.3 State user interface design activities.
7.4 List GUI implementation tools.
7.5 State user interface golden rules.

**8. Understand software design complexity**

8.1 Describe Halstead's complexity measures.
8.2 Describe Cyclomatic complexity measures.
8.3 State function point

**9. Understand software implementation**

9.1 Describe structured programming.
9.2 State functional programming.
9.3 State programming style and coding guideline.
9.4 Describe software documentation
9.5 State software implementation challenges.

**10. Understand software testing process**

10.1 Describe software validation and verification
10.2 State manual vs automated testing
10.3 Describe testing approaches
10.4 State testing levels
10.5 Describe testing documentation
10.6 State testing vs quality control & assurance and audit

**11. Understand software maintenance overview**

11.1 Describe types of maintenance
11.2 List cost of maintenance
11.3 State maintenance activities
11.4 State software re-engineering
11.5 Describe component reusability

**12. Understand Scrum agile method**

12.1 Describe scrum framework and sprints
12.2 Sate scrum roles
12.3 State scrum master roles
12.4 Describe scrum events (sprint, planning, daily scrum meeting, sprint review, retrospective)
12.5 State artifacts
12.6 State user stories
12.7 Describe burn down charts
12.8 State estimation process
12.9 State scrum tools and benefits

## Practical:

1 Measure the complexity of a given source code based on
    a. Halstead's Complexity Measures
    b. Measure cyclomatic complexity of a give code or software.
    c. Identify code blocks
    d. Draw Flow chart
    e. Draw flow graph
2 Measure function point of a given software.
3 Draw a data flow diagram from a given case study.
4 Draw structure chart form a given case study
5 Draw a HIPO diagram for a software requirement.
6 Do requirement analysis for a given case study and prepare requirement document
    a. Gather user requirement
    b. Write sample SRS
    c. Apply Requirement Elicitation Techniques to validate requirements
7 Identify Modules from a case study
    a. Identify Modules
    b. Identify sequential and concurrent units
8 Identify coupling and cohesion From a object oriented design
9 Write a function requirement on structured English model
10 Write pseudo – code of a given problem
11 Prepare a decision table from a given problem
12 Draw entity relationship model from a given case study.
13 Write a object oriented design from a given case study
    a. Write the objects, class
    b. Write Modules
    c. Draw object relationship diagram
14 Design a prototype implementation of a software using GUI
    a. Identify the GUI requirements
    b. List down application specific GUI requirements
    c. Draw a prototype implementation
    d. Draw a prototype design using GUI tools
15 Write a functional code for a given problem
    a. Functional programming approach
    b. Object oriented approach
16 Write a sample software following provided coding guideline
17 Write sample software documentation
    a. Requirement documentation
    b. Design documentation
    c. Technical documentation (code commenting and explanation)
    d. User documentation user guide

18 Write re-usable code or module
    a. Write sample library module
    b. Version control using tools (git, svn)
    c. Write machine independent code

19 Write Test documentation
    a. Write test case for a given problem
    b. Write Unit test cases
    c. Write Functional test cases
    d. Write user interface test cases
    e. Write a automated test program

20 Practice sample scrum using any open source tools
    a. Practice scrum events
    b. Prepare sample artifacts for a project
    c. Write user stories
    d. Prepare burn down chart
    e. Practice estimation planning poker

**REFERENCE BOOKS AND URL.**

1. Software engineering – A practitioner's approach - Mc GRAW – HILL by Roger S. Pressman
2. Introduction to system analysis and design – Prentice Hall by IgroHawryszkiewycz

Related URL links:

3. http://www.vumultan.com/Books/CS605-Software%20Engineering%20Practitioner%E2%80%99s%20Approach%20%20by%20Roger%20S.%20Pressman%20.pdf
4. https://www.tutorialspoint.com/software_engineering/index.htm
5. https://www.tutorialspoint.com/scrum/index.htm