

1

IT4492/IT4492E

Structured Programming

Lecturer:

Office:

Phone:

Email:

Web site:

Reception hours:

Nguyen Thanh Hung

R.601 B1 building
1A Dai Co Viet street
10000 Hanoi

(84 4) 3 8 68 25 95

hungnt@soict.hust.edu.vn

<http://www.soict.hust.edu.vn/~hungnt>

only by prior arrangement

@2017-2018

2

Information

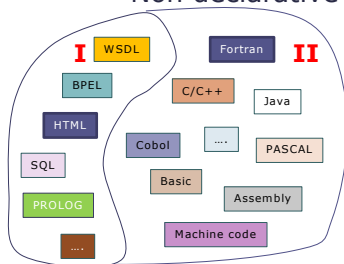
- Class: ICT – K60
 - Location: R.406, D9 building
 - Schedule: Thu. 14h15 - 15h50
(45 minutes x 2 x 15 weeks)

3

3

Introduction

- Given a problem, how to:
 - Design an algorithm for solving it
 - Implement this algorithm as a computer program
- Needs of programming languages and paradigms
- Language: express the algorithm to a machine
 - Declarative language (**I**): what to do, what to store
 - Non declarative language (**II**): how to do, how to store



4

4

Introduction

- Given a problem, how to:
 - Design an algorithm for solving it
 - Implement this algorithm as a computer program
- Needs of programming languages and paradigms
- Paradigm: comprise a set of concepts that are used as patterns for programming

First do this and next do that

Imperative

Evaluate an expression and use the resulting value for something

Functional

Answer a question via search for a solution

Logical

Send messages between objects to simulate the temporal evolution of a set of real world phenomena

Object-oriented

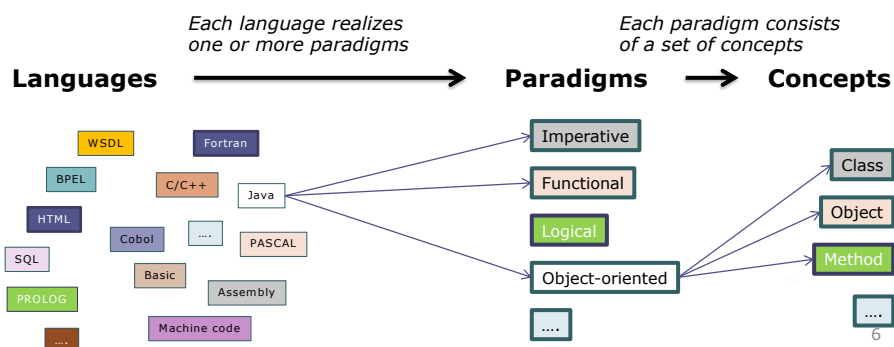
...

5


5

Introduction

- Given a problem, how to:
 - Design an algorithm for solving it
 - Implement this algorithm as a computer program
- Needs of programming languages and paradigms



6




Introduction

How do you organize your code ???

- From **bad programming habits** ...
 - Foggy idea about what is to be done
 - Write program with no planning: start from the beginning and write to the end
 - No systematic debugging: considered it finished if it works on only one test case
- ... to **undesirable results**
 - Do not know how to program
 - Any change of requirement invites rewriting of the entire program again
 - Program with numerous bugs that take extremely long time to debug, or even failure to complete

7

7



Introduction

How do you organize your code ???

- Structured programming
 - Becomes popular since the 70's
 - Should have been learnt by students that have taken any programming course
 - Absolutely essential for handling software projects

8

8

Course Description

- Structured programming in software projects
- Principles and patterns central to successful software development
- Modern development methods
 - **rapid software development**
 - **software reuse**
 - software as a service
- Well-structured software practices

9

9

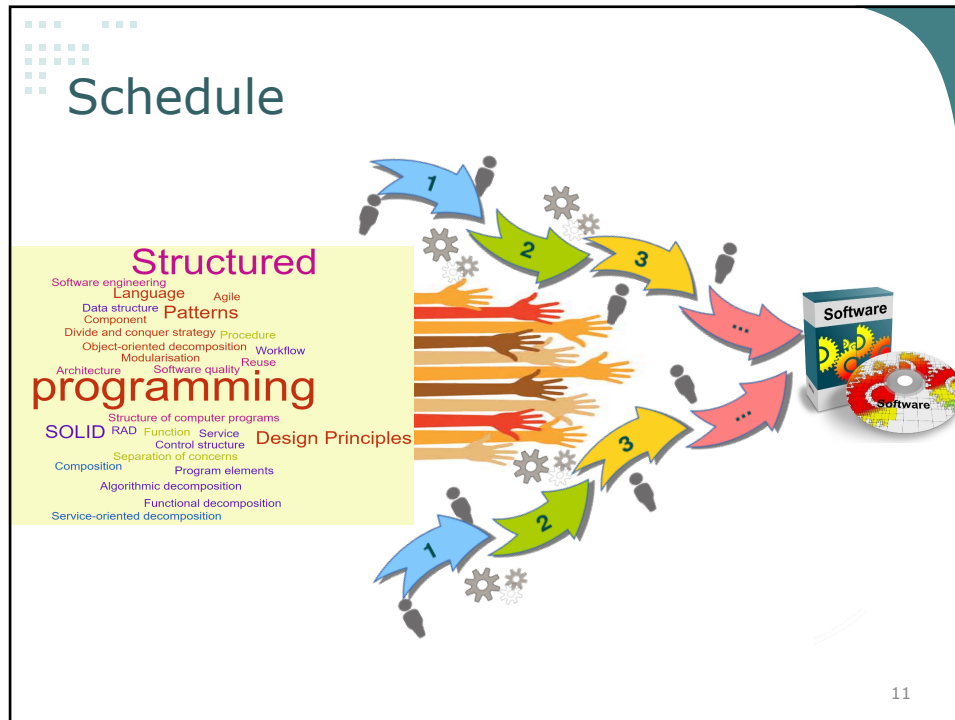
Objectives

Upon completion of this course, students will be able to:

- Knowledge:
 - Identify the needs for structured programming in software projects
 - Distinguish the program's low level structures that can be implemented by various programming languages
 - Summarize general design principles in software engineering and modern development methods
 - Determine good experiences as software patterns
- Skills:
 - Visualize the program structure in software design and development
 - Recognize, produce and maintain well-structured software
- Attitude:
 - Revise the program structure in software design and development
 - Show off a well-structured programming style

10

10



11

Books & Materials

- Required texts:
 - SE1: Software Engineering, Ian Sommerville, Addison-Wesley, ISBN 978-0-321-31379-9
 - SE2: Software Engineering: Theory and Practice, Shari Lawrence Pfleeger, Prentice-Hall, ISBN 0-13-624842-X
 - SE3: Software Engineering: A practitioner's approach, Roger Pressman, McGrawHill Education, ISBN 978-0-07-802212-8
 - FSE: Fundamentals of Software Engineering, Carlo Ghezzi, et al., Pearson Education, ISBN 0-13-305699-6
 - ASD: Agile Software Development: Principles, Patterns, and Practices, Robert C. Martin, Pearson Education, ISBN: 0-13-597444-5
 - PSOA & WS: Pattern Service-Oriented Architecture and Web Services, Mark Endrei et al., IBM red-books
- Additional texts:
 - DP: Design Patterns: Elements of Reusable Object-Oriented Software, Erich Gamma, et al., Addison-Wesley, ISBN: 0-201-633361-2
 - JDP: Java Design Patterns, James W. Cooper, Addison-Wesley, ISBN: 0-201-48539-7

18

18

