

# INTRODUCTION SOFTWARE ENGINEERING I



# GRADING POLICY (TENTATIVE)

Every student has A as of today

- Your future grade will be determined using the following criteria

## Theory

10% Sectional-1

15% Sectional-2

50% Final

15% Assignments & Project

10% Quizzes



# COURSE DETAILS

## ☐ Books

- ☒ *Somerville, Ian. Software Engineering, 9th Edition, Addison-Wesley, 2006*
- ☒ R.S. Pressman, *Software Engineering: A Practitioner's Approach*, 6th ed., McGraw-Hill Book Co., NY, 2005

## Policies

- ☒ Attendance policy
- ☒ Mobile phone usage
- ☒ Assignment submission
- ☒ Quizzes
- ☒ Fines
- ☒ Plagiarism policy

- ☐ Important Rule – **Don't Cheat!** (There will not be a 2<sup>nd</sup> warning)

# LECTURE OUTCOMES

At the end of this lecture you should be able to answer the following key questions:

- ⊠ What is software engineering?
- ⊠ Why is software engineering important?
- ⊠ What is the role of software in today's age?

# COURSE OUTLINE

Software Engineering I  
This Semester

**Requirements  
Analysis and  
Architecture**

Software Engineering II  
Next Semester

**Design and  
Implementation**

# WHAT IS (NOT) S.E.?

Not just software programming

☒ Individual vs. team

Not just a process

☒ Field that studies several different processes



# WHAT IS SOFTWARE ENGINEERING?

An Exercise: List all the ways in which

- ☒ Building a tinker-toy car for fun is different from designing a new automobile
- ☒ Building a tree-house is different from the architecture of a high-rise office building

So, how might software “engineering” be different from writing a 50-line Java applet?

# SOFTWARE VARIES

Size

How humans interact with it

Requirements stability/knowledge

Need for reliability

Need for security

Portability

Cost



# MICROSOFT POWERPOINT



Size: large

Interactiveness: high

Requirements: frequent new features

Reliability: moderate

Security: low (at least used to be)

Portability: high

Cost: high

# SPACE SHUTTLE SOFTWARE



Size: moderate to large

Interactiveness: low

Requirements: stable

Reliability: very high

Security: low

Portability: low

Cost: high

# ONLINE SHOPPING SOFTWARE



Size: moderate

Interactiveness: high

Requirements: frequent new features

Reliability: moderate

Security: high

Portability: low

Cost: low

# YOUR EXAMPLE

Size:

Interactiveness:

Requirements:

Reliability:

Security:

Portability:

Cost:

# SOFTWARE ENGINEERING

## What is Software?

- ⊠ *The programs, routines, and symbolic languages that control the functioning of the hardware and direct its operation.*
- ⊠ *A set of instructions that cause a computer to perform one or more tasks. The set of instructions is often called a program*

# SOFTWARE ENGINEERING

## What is Engineering?

⌘ *Engineering is the discipline and profession of applying technical, scientific and mathematical knowledge in order to utilize natural laws and physical resources to help design and implement materials, structures, machines, devices, systems, and processes that safely realize a desired objective.*

# SOFTWARE ENGINEERING

What is Software Engineering?

- ⊠ *Software engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software.*
- ⊠ *Software engineering encompasses a process, management techniques, technical methods, and the use of tools.*

# WHY LEARN SOFTWARE ENGINEERING?

Not necessarily learning software engineering to engineer software.

By the end of this course, you will be trained on how to carry out a project successfully, how to communicate with the stakeholders & how to transform this communication into powerful models which will lead to a winning product.

Moreover you will also learn how to test these models & how to evaluate risks at every stage of the development process.



# WHAT WILL YOU LEARN?

1. Understanding, clarifying and documenting the **requirements** for a customer
2. **Planning** how to conduct a project
3. **Designing** a software system at the architectural and detailed levels
4. Reviewing and **testing** designs and implementations

# SOFTWARE PRODUCTS (TYPES)

## Generic products

- ☒ Stand-alone systems that are marketed and sold to any customer who wishes to buy them.
- ☒ Examples – PC software such as graphics programs, project management tools; CAD software; software for specific markets such as appointments systems for dentists.

## Customized products

- ☒ Software that is commissioned by a specific customer to meet their own needs.
- ☒ Examples – embedded control systems, air traffic control software, traffic monitoring systems.

# FREQUENTLY ASKED QUESTIONS ABOUT SOFTWARE ENGINEERING

Question	Answer
What is software?	Computer programs and associated documentation. Software products may be developed for a particular customer or may be developed for a general market.
What are the attributes of good software?	Good software should deliver the required functionality and performance to the user and should be maintainable, dependable and usable.
What is software engineering?	Software engineering is an engineering discipline that is concerned with all aspects of software production.
What are the fundamental software engineering activities?	Software specification, software development, software validation and software evolution.
What is the difference between software engineering and computer science?	Computer science focuses on theory and fundamentals; software engineering is concerned with the practicalities of developing and delivering useful software.
What is the difference between software engineering and system engineering?	System engineering is concerned with all aspects of computer-based systems development including hardware, software and process engineering. Software engineering is part of this more general process.

# FREQUENTLY ASKED QUESTIONS ABOUT SOFTWARE ENGINEERING

Question	Answer
What are the key challenges facing software engineering?	Coping with increasing diversity, demands for reduced delivery times and developing trustworthy software.
What are the costs of software engineering?	Roughly 60% of software costs are development costs, 40% are testing costs. For custom software, evolution costs often exceed development costs.
What are the best software engineering techniques and methods?	While all software projects have to be professionally managed and developed, different techniques are appropriate for different types of system. For example, games should always be developed using a series of prototypes whereas safety critical control systems require a complete and analyzable specification to be developed. You can't, therefore, say that one method is better than another.
What differences has the web made to software engineering?	The web has led to the availability of software services and the possibility of developing highly distributed service-based systems. Web-based systems development has led to important advances in programming languages and software reuse.

# ESSENTIAL ATTRIBUTES OF GOOD SOFTWARE

Product characteristic	Description
Maintainability	Software should be written in such a way so that it can evolve to meet the changing needs of customers. This is a critical attribute because software change is an inevitable requirement of a changing business environment.
Dependability and security	Software dependability includes a range of characteristics including reliability, security and safety. Dependable software should not cause physical or economic damage in the event of system failure. Malicious users should not be able to access or damage the system.
Efficiency	Software should not make wasteful use of system resources such as memory and processor cycles. Efficiency therefore includes responsiveness, processing time, memory utilisation, etc.
Acceptability	Software must be acceptable to the type of users for which it is designed. This means that it must be understandable, usable and compatible with other systems that they use.

# APPLICATION TYPES

## Data collection systems

- ⊠ These are systems that collect data from their environment using a set of sensors and send that data to other systems for processing.

## Systems of systems

- ⊠ These are systems that are composed of a number of other software systems.

# SOFTWARE ENGINEERING AND THE WEB

The Web is now a platform for running application and organizations are increasingly developing web-based systems rather than local systems.

Web services allow application functionality to be accessed over the web.

Cloud computing is an approach to the provision of computer services where applications run remotely on the 'cloud'.

☒ Users do not buy software but pay according to use.

# HARDWARE VS. SOFTWARE

## Hardware

- Manufactured
- Wears out
- Built using components
- Relatively simple

## Software

- Developed/Engineered
- Deteriorates
- Custom built
- Complex



# MANUFACTURING VS. DEVELOPMENT

Once a hardware product has been manufactured, it is difficult or impossible to modify. In contrast, software products are routinely modified and upgraded.

In hardware, hiring more people allows you to accomplish more work, but the same does not necessarily hold true in software engineering.

Unlike hardware, software costs are concentrated in design rather than production.

# SOFTWARE MYTHS

If we get behind schedule, we can add more programmers and catch up.

A general statement about objectives is sufficient to begin building programs.

Change in project requirements can be easily accommodated because software is flexible.

# CUSTOMER MYTHS



“A general statement of objectives is sufficient to begin writing programs - we can fill in the details later ...”

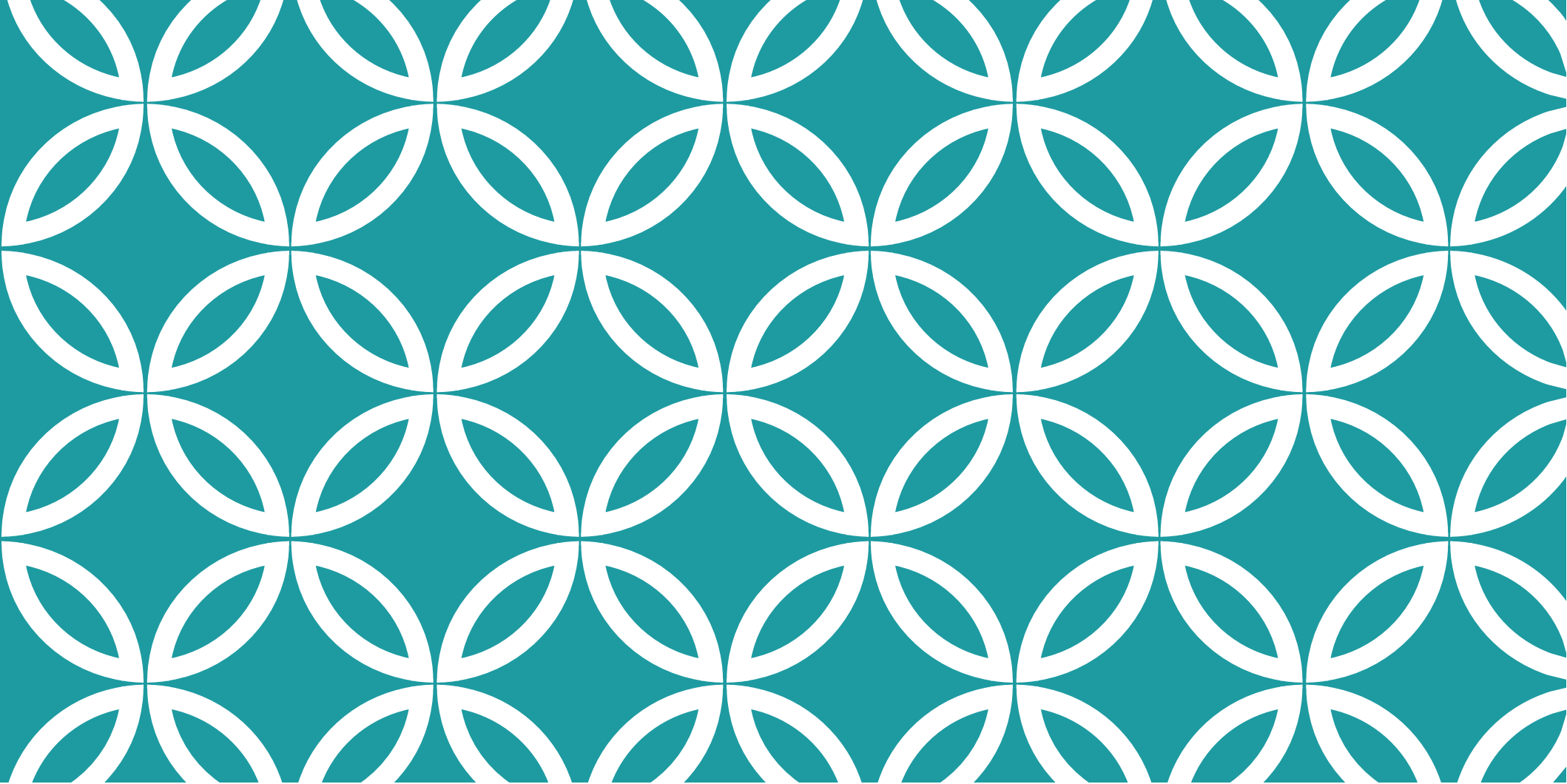
“Project requirements continually change but this change can easily be accommodated because software is flexible ...”

# ASSIGNMENTS

Write down brief description of few software applications that are being used at various organizations in Pakistan

(Deadline – 18<sup>th</sup> Feb)

- Read Chapter 1 and be prepare for quiz



QUESTIONS?