**Lecture Objectives**

- To define the software engineering process
- To understand the importance of software engineering
- To discuss the important characteristics of software
- To understand that the quality of different applications may be evaluated differently

**Typical Approaches**

- Go to the computer and immediately write the program
- Find an old program and modify it
- Discuss with friends on how to do it
- Ask the lecturer for more information about the program

**Course Materials**

- Intranet: mmlscyber

**A simple program**

"Write a program to get a list of students’ test marks, calculate the grades and print a report of the results”

**What is Software Engineering**

- Software
  - programs that provide function & performance
  - data structures for information manipulation
  - documents that describe the operations and use of the programs
- Engineering
  - A discipline that applies scientific and technical methods in the design and production of a product
**Definition of Software Engineering**

IEEE Definition:
The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software.

**Another Definition of Software Engineering**

The practical application of scientific knowledge in the design and construction of computer programs and the associated documentation required to develop, operate, and maintain them. (Boehm).

**Objectives of Software Engineering**

- To improve quality of software products
- To increase customer satisfaction
- To increase productivity
- To increase job satisfaction

Software engineering is not programming. Programming is an important part of software engineering. "This is not a programming course.”

**Historical Background**

- Early days of computing, programs were written to make hardware work
- Programming was not a discipline, more like a hobby or "art form" 
- However, computer developments require larger programs to be developed e.g. compilers and operating systems
- Programming becomes a profession.

**Programs’ characteristics**

- Programs before were small done by one expert person (programming is small) for one algorithm. The input was a numerical data and the output was outputted to a printer. Trouble shooting was done at the memory and registers level.
- Programs now are large and complex written by groups of people (programming in large).

**Computer Expenditure**

- Total cost

  - Hardware Maintenance
  - Software Development
  - Software Maintenance

  1955 - 1980s
Software Crisis

- The large programming projects required many programmers working together
- The projects were not delivered on time and costs more than initial budget - software crisis
- Software Engineering methods were developed to overcome these problems

Software Characteristics

- Software is developed or engineered, not manufactured in the classical sense
- Software doesn’t “wear out”
- Most software is custom-built, rather than being assembled from existing components

What Is A Good Software?

- Software is intangible
- Good software is subjective
- Some qualities that are used to assess software:
  - Correctness: a program satisfies its specifications.
  - Reliability: a program satisfies its intended functions.
  - Usability: the effort required to learn, operate, prepare input, and interpret the output.
  - Integrity: Control of access to unauthorized persons.

What Is A Good Software continue?

- Efficiency: amount of computing resources required.
- Maintainability: effort required to locate and fix errors in an operational programs.
- Portability: required effort to transfer a program from one hardware/software environment to another.
- Testability: required effort to test a program to ensure its performing its intended functions.
- Interoperability: effort required to couple programs.
- Reusability: reuse of programs in other applications.

Software Applications

- System Software
- Real-time Software
- Business Software
- Engineering & Scientific Software
- Embedded Software
- Personal Computer Software
- Artificial Intelligence Software
References

END