



Computer Fundamentals

Dr. Safdar Nawaz Khan Marwat
DCSE, UET Peshawar

Lecture 18



Outline

- Creating computer programs



Objectives

- Define term computer program
- Describe use of flowcharts and pseudocode in programming
- Identify ways in which a program can work toward a solution
- Object Oriented Programming



What is a Computer Program?

- Computer program
 - ❑ Also called software
 - ❑ A list of instructions
 - ❑ Instructions are called code
 - ❑ CPU performs instructions
 - ❑ Types
 - System software
 - Application



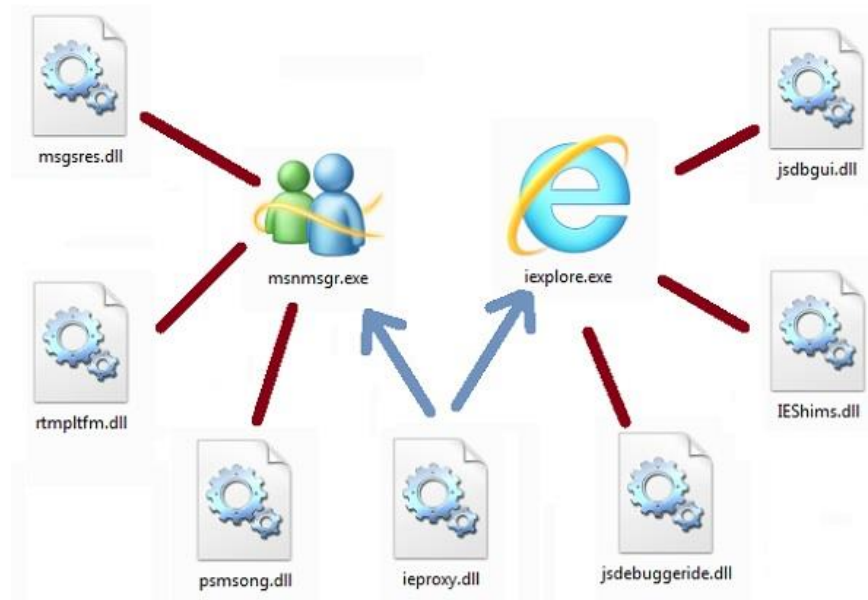
Software is Stored in Many Files

- Executable files
 - ❑ Contain instructions for CPU
 - ❑ Have extensions of .exe, or .com



Software is Stored in Many Files (cont.)

- Dynamic link libraries
 - ❑ Partial executable file
 - ❑ Used to support executable files
 - ❑ Have .dll extensions
 - ❑ Several .exe files can use a single .dll file



Source: <https://www.maketecheasier.com/what-are-dll-files>



Software is Stored in Many Files (cont.)

➤ Initialization files

- ❑ Contain configuration settings for software
 - E.g. size and starting point of window
 - E.g. background color etc.
- ❑ Have a .ini extension
- ❑ Modern programs use Windows registry
 - Special database for holding user info



Software is Stored in Many Files (cont.)

➤ Help files

- ☐ Contain information about software
- ☐ Information is indexed and searchable
- ☐ Provides an online manual
- ☐ Have a .chm or .hlp extension

➤ Batch files

- ☐ Contain sequence of commands for OS
- ☐ Used to automate repetitive tasks
 - Created for command sequences which are repeatedly needed
- ☐ Text files with series of OS commands
- ☐ Have a .bat extension



Hardware/Software Interaction

➤ Program execution

- ❑ Software executes at CPU level
- ❑ Code to play a sound
 - Code generates an interrupt
 - CPU tells the sound card to play
 - Sound card plays the file
- ❑ Programmer creates code



Hardware/Software Interaction (cont.)

➤ Code

- ❑ Statements written in a programming language
- ❑ Writing code can be tedious
 - Code must be perfect
 - Order of steps must be exact
- ❑ Writing code is quite exciting
 - Problems are solved
 - New ideas are formed



Hardware/Software Interaction (cont.)

➤ Writing code

The screenshot shows the Microsoft Visual C++ 6.0 IDE. The title bar reads "13a18 - Microsoft Visual C++ - [13a18.cpp]". The menu bar includes File, Edit, View, Insert, Project, Build, Tools, Window, and Help. The toolbar contains icons for file operations and a search box with the text "account". The left sidebar shows a project explorer with "13a18 classes" and tabs for "ClassView" and "FileView". The main editor window displays the following C++ code:

```
float average;  
getAges(age1, age2, age3);  
average = averageAges(age1, age2, age3);  
cout << The average age is "<< average << endl;  
}
```

Below the editor, the "Output" window shows the compilation process and several errors:

```
Compiling...  
13a18.cpp  
D:\My Documents\!books\norton intro\Unit 13 original\13a18.cpp(10) : error C2065:  
D:\My Documents\!books\norton intro\Unit 13 original\13a18.cpp(10) : error C2146:  
D:\My Documents\!books\norton intro\Unit 13 original\13a18.cpp(10) : error C2146:  
D:\My Documents\!books\norton intro\Unit 13 original\13a18.cpp(10) : error C2001:  
D:\My Documents\!books\norton intro\Unit 13 original\13a18.cpp(10) : error C2065:  
D:\My Documents\!books\norton intro\Unit 13 original\13a18.cpp(10) : error C2146:  
D:\My Documents\!books\norton intro\Unit 13 original\13a18.cpp(10) : error C2065:  
D:\My Documents\!books\norton intro\Unit 13 original\13a18.cpp(10) : error C2143:  
D:\My Documents\!books\norton intro\Unit 13 original\13a18.cpp(11) : error C2143:  
Error executing cl.exe.
```

The status bar at the bottom indicates "D:\MY DOCUMENTS\!BOOKS\NORTON INTRO\UNIT 13 ORIGINAL\13a18.cpp saved" and "Ln 10, Col 1".



Hardware/Software Interaction (cont.)

➤ Machine code

- ❑ Computers operate in binary
- ❑ Code is translated into machine code
 - CPU executes the machine code
- ❑ CPUs have a unique machine code
- ❑ Machine language is too obscure
 - Complex for using in software development

Machine Code

000101001011010101010101
1110110101010101010101100
001010010101001011101011
100101001011010101010101
011010010011001011101011
0001000101011010101010001
1010100101010010101101011
000101001011010101010101



Hardware/Software Interaction (cont.)

- Assembly language
 - ❑ Low-level language
 - ❑ Designed for a specific family of processors
 - ❑ Represents various instructions in symbolic code
 - ❑ More understandable form
 - ❑ Assembly language converted into executable machine code by a utility program referred to as an assembler

Assembly Code

```
;CLEAR SCREEN USING BIOS
CLR: MOV AX,0600H      ;SCROLL SCREEN
    MOV BH,30          ;COLOUR
    MOV CX,0000        ;FROM
    MOV DX,184FH       ;TO 24,79
    INT 10H            ;CALL BIOS;

;INPUTTING OF A STRING
KEY: MOV AH,0AH        ;INPUT REQUEST
    LEA DX,BUFFER      ;POINT TO BUFFER WHERE STRING STORED
    INT 21H            ;CALL DOS
    RET               ;RETURN FROM SUBROUTINE TO MAIN PROGRAM;

; DISPLAY STRING TO SCREEN
SCR: MOV AH,09         ;DISPLAY REQUEST
    LEA DX,STRING      ;POINT TO STRING
    INT 21H            ;CALL DOS
    RET               ;RETURN FROM THIS SUBROUTINE;
```



Hardware/Software Interaction (cont.)

- Programming languages
 - ❑ Simplifies the writing of code
 - English is used to describe the binary
 - ❑ Original code is called source code
 - ❑ Several hundred languages exist



Hardware/Software Interaction (cont.)

- Compilers and interpreters
 - ❑ Converts source code into binary
 - Allows code to execute
 - ❑ Checks source code for correctness



Hardware/Software Interaction (cont.)

➤ Compiler

- ❑ Covert source code to machine code all at once
- ❑ Creates an executable file
 - Compiler output contents are called object code
- ❑ Executable can run on its own
- ❑ Each language has its own compiler
- ❑ C++ and Java are compiled languages



Hardware/Software Interaction (cont.)

➤ Interpreter

- ❑ Runs program one line at a time
- ❑ More flexible than compilers
 - Translates code on the fly
- ❑ Slower than compilers
- ❑ Visual Basic and Perl are interpreted languages