Computing System

Definitions:

1. Computer Science – Study of algorithms and their efficient implementation in a computer.
2. Computer System – Used to solve problems and interact with its environment.
3. Computer – Programmable electronic device that can STORE, RETRIEVE, and PROCESS data.
4. Computer Hardware - Collection of physical elements that make up the machine and its related pieces
5. Computer Software - Collection of programs that provide the instructions that a computer carries out
6. Von Neumann architecture

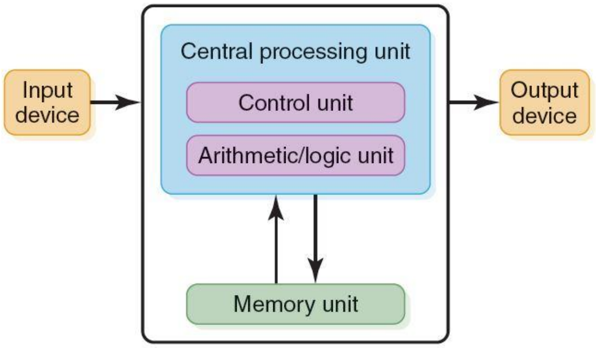


FIGURE 5.1 The von Neumann architecture.

7. Bits – Short for binary digit (0 or 1)
8. Characters – Decimal digits, letters & special symbols
   a. Unicode – composed of 8, 16 or 42 bits
   b. ASCII – American Standard Code for Information Interchange
9. Fields – Composed of characters or bytes that conveys meaning
10. Records – Several related fields
11. Files – Group of related records; sequence of bytes
12. Database – Collection of data organized for easy access & manipulation
    a. Relational Database
13. Big Data – Applications deal with massive amounts of data
14. Machine Language – Language made up of binary-coded instructions that is used directly by the computer.
15. Assembly Language – Low-level programming language in which a mnemonic represents each of the machine-language instructions for a particular computer
16. Assembler – A program that translates an assembly-language program in machine code
17. Assembler Directive – Instructions to the translating program (assembler)
18. Programming Language – A set of rules, symbols, and special words used to construct programs
19. Syntax Rules – Rules that tell you which statements (instructions) are legal, or accepted by the programming language
20. Semantic Rules – Rules that determine the meaning of the instructions
21. Comments - Explanatory text for the human reader
    a. Single-Line – One line only //
    b. Multi-Line – Multiple lines starting with /* and ending with */
22. Identifier – Names of things such as variables, constants, and methods; Predefined or user defined
    a. Must consist of letters, digits, Underscore (_) or dollar sign ($)
    b. Must begin with letter, underscore or dollar sign.
23. Datatype – Set of values together with a set of operations on those values

24. Primitive Datatypes – Fundamental data types in Java
    a. Integral – Deals with integers, numbers without a decimal and characters
        i. Char – represents single characters
        ii. Byte – 8 bits
        iii. Short – 16 bits
        iv. Int – Just like integers in math; positive & negative; no commas
        v. Long – 64 bits
    b. Floating-Point – Deals with decimal numbers; tend to be larger numbers
        i. Double – larger decimal numbers
    c. Boolean – Logical values; True or False
25. Java Application – A computer program that executes when you use the java command to launch the Java Virtual Machine (JVM)
26. Class Declaration – Every Java program consists of at least one class that you define
    a. Class keyword introduces a class declaration followed by the class name
27. Filename for a public Class – A public class must be placed in a file that has a file name of the form ClassName.java.
28. Class Names & Identifiers – Class Names are capitalized; Variables are not
29. Class Body – { contained between curly braces }

    ```
    public class ClassName {

    }
    ```

30. main Method –

    ```
    public static void main(String[] args) {


    }
    ```

31. Method Body – { contained between curly braces }; code goes inside
32. System.out print methods (what are they and what do they do?)

    a. print() – prints a line
    b. println() – prints a line and moves the curser to the next line
    c. printf() – formats what needs to be printed using specifiers
33. Escape Character (what is it and list some examples) \
    a. \n
    b. \”
34. Expressions – portions of statements that contain calculations
35. Arithmetic operators
    a. +
    b. -
    c. *
    d. /
    e. %
36. Assignment Operator
    - =
    - works right to left

37. Equality Operators
    a. ==
    b. !=
38. Relational Operators
    a. <
    b. >
    c. <=
    d. >=
39. If statements –

    <u>if</u>

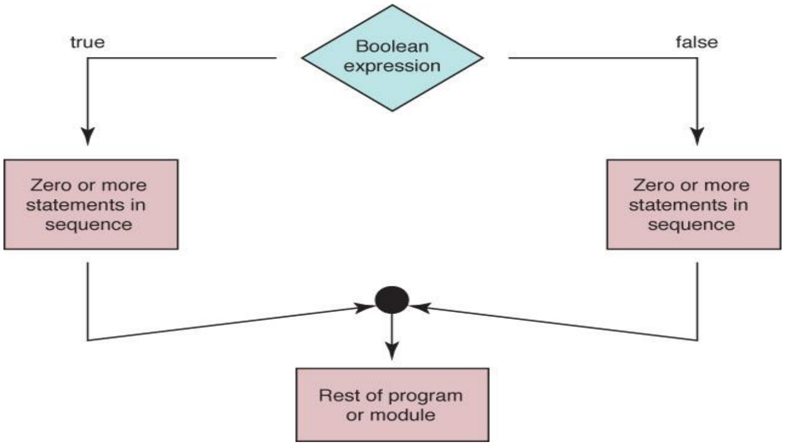
    Performs an action, if a condition is true; skips it, if false.

    Single-selection statement—selects or ignores a single action (or group of actions).

    <u>if…else</u>

    Performs an action if a condition is true and performs a different action if the condition is false.

    Double-selection statement—selects between two different actions (or groups of actions).
40. Condition – Boolean expression used in selection or iteration statements
41. IF Statement Control Flow Diagram


true
Boolean expression
false
Zero or more statements in sequence
Zero or more statements in sequence
Rest of program or module

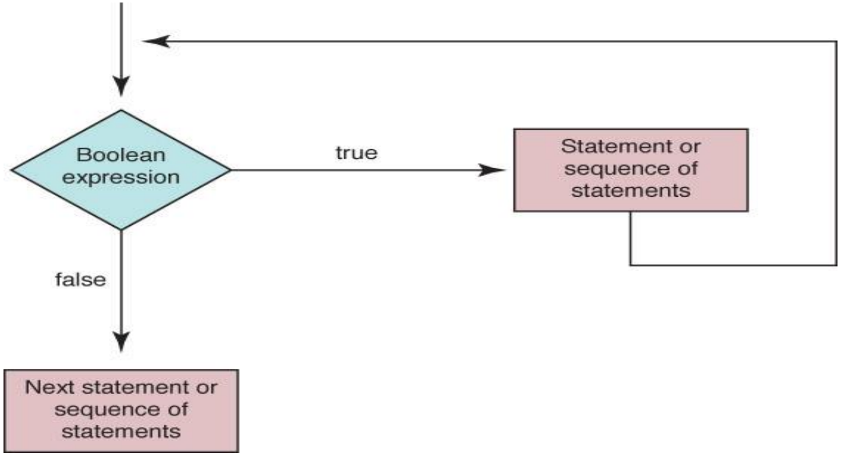
42. String
    - ○ Strings in Java are enclosed in Double quotation marks
    - ○ Characters are enclosed in Single quotation marks.
    - ○ The quotation marks and the characters between them are a string
43. Algorithm
    - ○ An algorithm is a procedure for solving a problem in terms of
        - ▪ the actions to execute and
        - ▪ the order in which these actions execute

44. Pseudocode – An algorithm is a procedure for solving a problem in terms of the actions to execute and the order in which these actions execute
45. Sequential execution – An algorithm is a procedure for solving a problem in terms of the actions to execute and the order in which these actions execute

46. Selection statements (3 types)
    a. An algorithm is a procedure for solving a problem in terms of
    b. the actions to execute and
    c. the order in which these actions execute

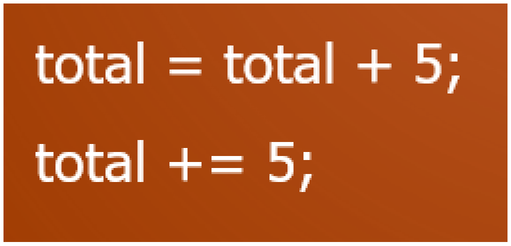
47. Iteration statements – Perform statements repeatedly while a loop-continuation condition remains true.

48. Looping Statement Control Flow Diagram


Boolean expression
true
Statement or sequence of statements
false
Next statement or sequence of statements

49. while Loops
    a. Counter controlled
    b. Sentinel controlled
50. Compound Assignment Operators

    += -= *= /= %=

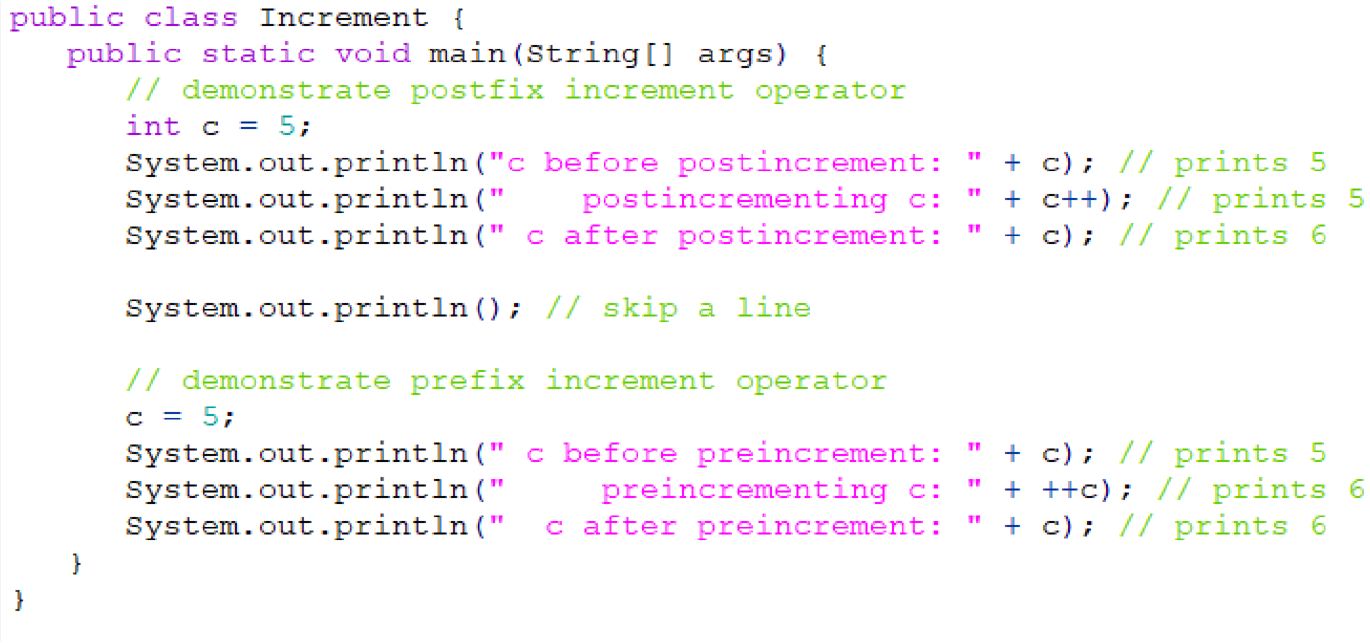


```
total = total + 5;

total += 5;
```

51. Increment & Decrement Operators

    ++ --



```
public class Increment {
   public static void main(String[] args) {
      // demonstrate postfix increment operator
      int c = 5;
      System.out.println("c before postincrement: " + c); // prints 5
      System.out.println("    postincrementing c: " + c++); // prints 5
      System.out.println(" c after postincrement: " + c); // prints 6

      System.out.println(); // skip a line

      // demonstrate prefix increment operator
      c = 5;
      System.out.println(" c before preincrement: " + c); // prints 5
      System.out.println("     preincrementing c: " + ++c); // prints 6
      System.out.println("  c after preincrement: " + c); // prints 6
   }
}
```

History

1. Abacus
    a. ~~16th Century~~ Ancient
    b. Instrument to record numeric values and on which a human can perform basic arithmetic
2. Blaise Pascal
    a. 17th Century
    b. Mechanical device to add and subtract
3. Leibniz Machine
    a. Gottfried Wilhelm von Liebniz
    b. 17th Century
    c. Calculating Machine, added function of multiplication and division
4. Jacquard's Loom
    a. 18th Century
    b. Joseph Jacquard
    c. Used for weaving cloth
    d. Series of cards with holes punched in to specify the use of specific colored thread to dictate the design that was woven
    e. First to use INPUT! Punched card
5. Analytical Machine
    a. 19th Century
    b. Charles Babbage
    c. Too complex for him to actually build with the technology of his time
    d. Never Implemented
    e. First design to include memory
6. Ada Augusta, Countess of Lovelace
    a. 19th Century
    b. Daughter of Lord Byron (English Poet)
    c. Extended Babbage's ideas and even corrected errors
    d. Credited at the first programmer
    e. Developed concept of a loop
        i. Series of repeated instructions
    f. Ada – programming language used by the DoD
7. Herman Hollerith
    a. 19th Century
    b. Developed first electro-mechancial tabulator
        i. Read info from punched cards
        ii. Revolutionized the census
    c. Later formed IBM (International Business Machines)
8. Alan Turing
    a. 1936
    b. British Mathematician
    c. Turing Machine
    d. Turing Test – Detects if AI is human like
9. Harvard Mark I
    a. 1944
    b. IBM automatic sequence controlled calculator given to Harvard
10. ENIAC
    a. 1946
    b. Electronic instead of electromechanical
    c. 1000 times faster than any previous computer

11. UNIVAC I
    - a. 1951
    - b. First commercial computer
    - c. First computer used to predict outcome of a presidential election
12. First Generation (1951 – 1959)
    - a. Hardware
        - i. Vacuum Tubes
        - ii. Magneitc Drum
        - iii. Card Reader
    - b. Software
        - i. Binary
        - ii. Assembly Languages
        - iii. Translators
        - iv. Programmers
            1. Application
            2. System
13. Second Generation (1959 – 1965)
    - a. Hardware
        - i. Transistor
        - ii. Immediate-Access Memory
            1. Magnetic cores
        - iii. Magnetic Disk
    - b. Software
        - i. High Level Languages
            1. FORTRAN
            2. COBOL
            3. Lisp
14. Moore's Law - From the invention of the integrated circuit, the number of circuits that could be placed on a single integrated circuit doubled each year.
15. Third Generation (1965 – 1971)
    - a. Hardware
        - i. Moore's Law
        - ii. Terminal
    - b. Software
        - i. Operating System
        - ii. Separation between users and hardware
            1. Computer programmers write programs to be used by general public (i.E., Nonprogrammers)
16. Hardware Fourth Generation (1971 – 1980?)
    - a. Large Scale Integration
    - b. Moore's Law Modified
        - i. Chip density doubling every 18 months
    - c. Personal Computer (PC)
17. Software Fourth Generation (1971 – 1989)
    - a. Structured Programming
        - i. PASCAL
        - ii. C++
    - b. Better Operating Systems
    - c. Applications
18. Hardware Fifth Generation

a. 1981 – IBM PC
b. 1984 – Apple MacIntosh
c. Workstations
  i. Networked
d. UNIX

19. Fifth Generation Software
  a. Rise of Microsoft
  b. Object-Oriented Design and Programming
  c. World Wide Web

Users

1) Programmers – solve specific problems
2) Systems Programmers – write more complex tools for other programmers
3) Applications Programmers – using complex tools to write programs for non-programmers to use
4) Everyone

Helpful Graphics

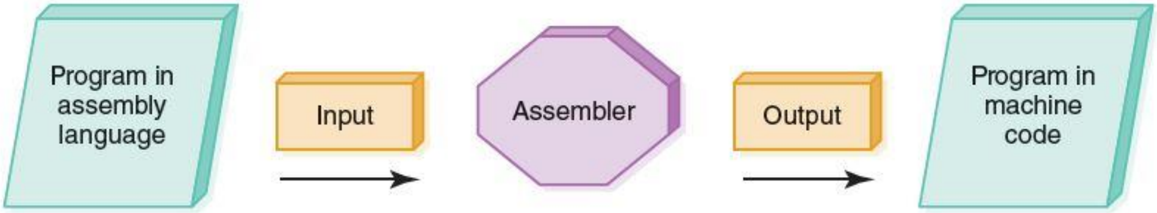
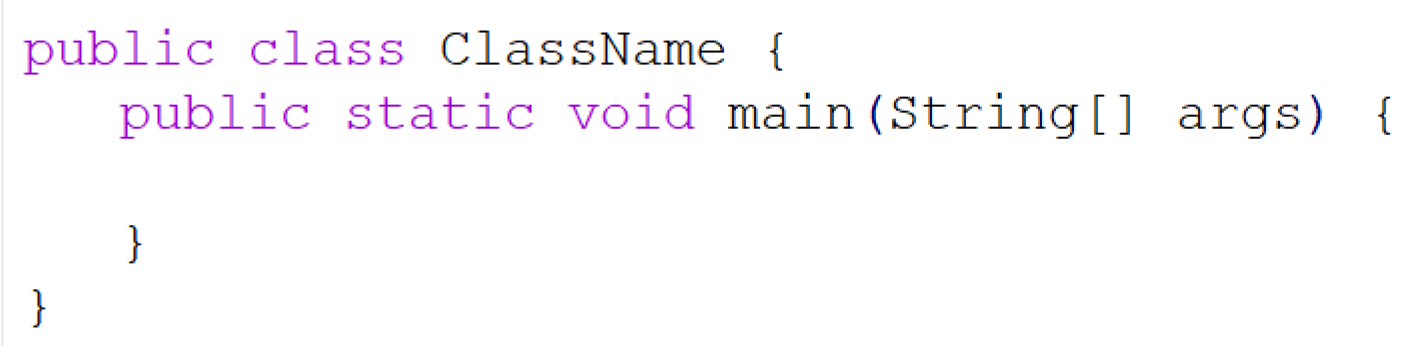


FIGURE 6.5 Assembly process



```
public class ClassName {
    public static void main(String[] args) {

    }
}
```