Advanced Computer Programming

[Lecture 1]

Saeed Reza Kheradpisheh

kheradpisheh@ut.ac.ir

Department of Computer Science
Shahid Beheshti University
Spring 1397-98

Algorithm

Definition

An **algorithm** is a process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer.

Definition

Algorithm design is the act of designing and describing the steps that are necessary for finding the solution for a problem.

- If you can't give written instructions for someone to solve the problem, there is no way the computer can magically find the right solution.
- The computer can only do **what you tell it to do**. It just does it faster, without getting bored or exhausted.

Algorithm Design: A Simple Example

Problem

You put $10,000 into a bank account that earns 5 percent interest per year. How many years does it take for the account balance to be double the original?

Solving by hand:

year	interest	balance
0		10000
1	10000.00 x 0.05 = 500.00	10000.00 + 500.00 = 10500.00
2	10500.00 x 0.05 = 525.00	10500.00 + 525.00 = 11025.00
3	11025.00 x 0.05 = 551.25	11025.00 + 551.25 = 11576.25
4	11576.25 x 0.05 = 578.81	11576.25 + 578.81 = 12155.06

Algorithm Design: A Simple Example

Problem

You put $10,000 into a bank account that earns 5 percent interest per year. How many years does it take for the account balance to be double the original?

Solving by hand:

year	interest	balance
0		10000
1	10000.00 x 0.05 = 500.00	10000.00 + 500.00 = 10500.00
2	10500.00 x 0.05 = 525.00	10500.00 + 525.00 = 11025.00
3	11025.00 x 0.05 = 551.25	11025.00 + 551.25 = 11576.25
4	11576.25 x 0.05 = 578.81	11576.25 + 578.81 = 12155.06

Keep computing until $20,000, Boring!

Algorithm Design: A Simple Example

- Computers are very good at carrying out repetitive calculations quickly and flawlessly.
- What is important to the computer is a description of the steps for finding the solution.
  - **Clear**, **unambiguous**, and requiring **no guesswork**.

Algorithm:

1. Start with a year value of **0**, a column for the interest, and a balance of **$10,000**.
2. **Repeat** the following steps while the balance is less than **$20,000**
   1. Add **1** to the year value.
   2. Compute the interest as balance x **0.05**.
   3. Add the interest to the balance.
3. Report the final year value as the answer.

Pseudocode

Definition

Informal description of an algorithm is called **pseudocode**.

- There are no strict requirements for pseudocode because it is read by human readers, not a computer program.
- Use statements such as the following to describe how a value is set or changed:
  - total cost = purchase price + operating cost
  - Multiply the balance value by 1.05.
  - Remove the first and last character from the word.

Pseudocode

Definition

Informal description of an algorithm is called **pseudocode**.

- There are no strict requirements for pseudocode because it is read by human readers, not a computer program.
- Use statements such as the following to describe how a value is set or changed:
  - total cost = purchase price + operating cost
  - Multiply the balance value by 1.05.
  - Remove the first and last character from the word.
- You can describe decisions and repetitions as follows:
  - If total cost 1 ¡ total cost 2
  - While the balance is less than $20,000
  - For each picture in the sequence

Pseudocode

- Use indentation to indicate which statements should be selected or repeated:
  - For each car

    operating cost = 10 x annual fuel cost

    total cost = purchase price + operating cost

Pseudocode

- Use indentation to indicate which statements should be selected or repeated:
  - For each car

    operating cost = 10 x annual fuel cost

    total cost = purchase price + operating cost
- Indicate results with statements such as:
  - Choose car1.
  - Report the final year value as the answer.

Pseudocode

- Use indentation to indicate which statements should be selected or repeated:
  - For each car
    operating cost = 10 x annual fuel cost
    total cost = purchase price + operating cost
- Indicate results with statements such as:
  - Choose car1.
  - Report the final year value as the answer.

Note

The exact wording is not important. What is important is that pseudocode describes a sequence of steps that is:

- Unambiguous
- Executable
- Terminating

Pseudocode

Definition

The step sequence is **unambiguous** when there are precise instructions for what to do at each step and where to go next.

Definition

A step is **executable** when it can be carried out in practice.

Definition

A sequence of steps is **terminating** if it will eventually come to an end.

Pseudocode

Definition

The step sequence is **unambiguous** when there are precise instructions for what to do at each step and where to go next.

Definition

A step is **executable** when it can be carried out in practice.

Definition

A sequence of steps is **terminating** if it will eventually come to an end.

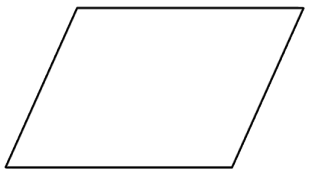
Definition

A sequence of steps that is unambiguous, executable, and terminating is called an **algorithm**.

Flowchart

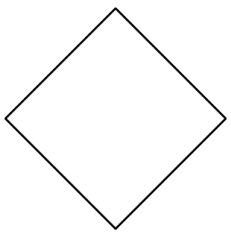
Beginning/ End

Direction of logic flow



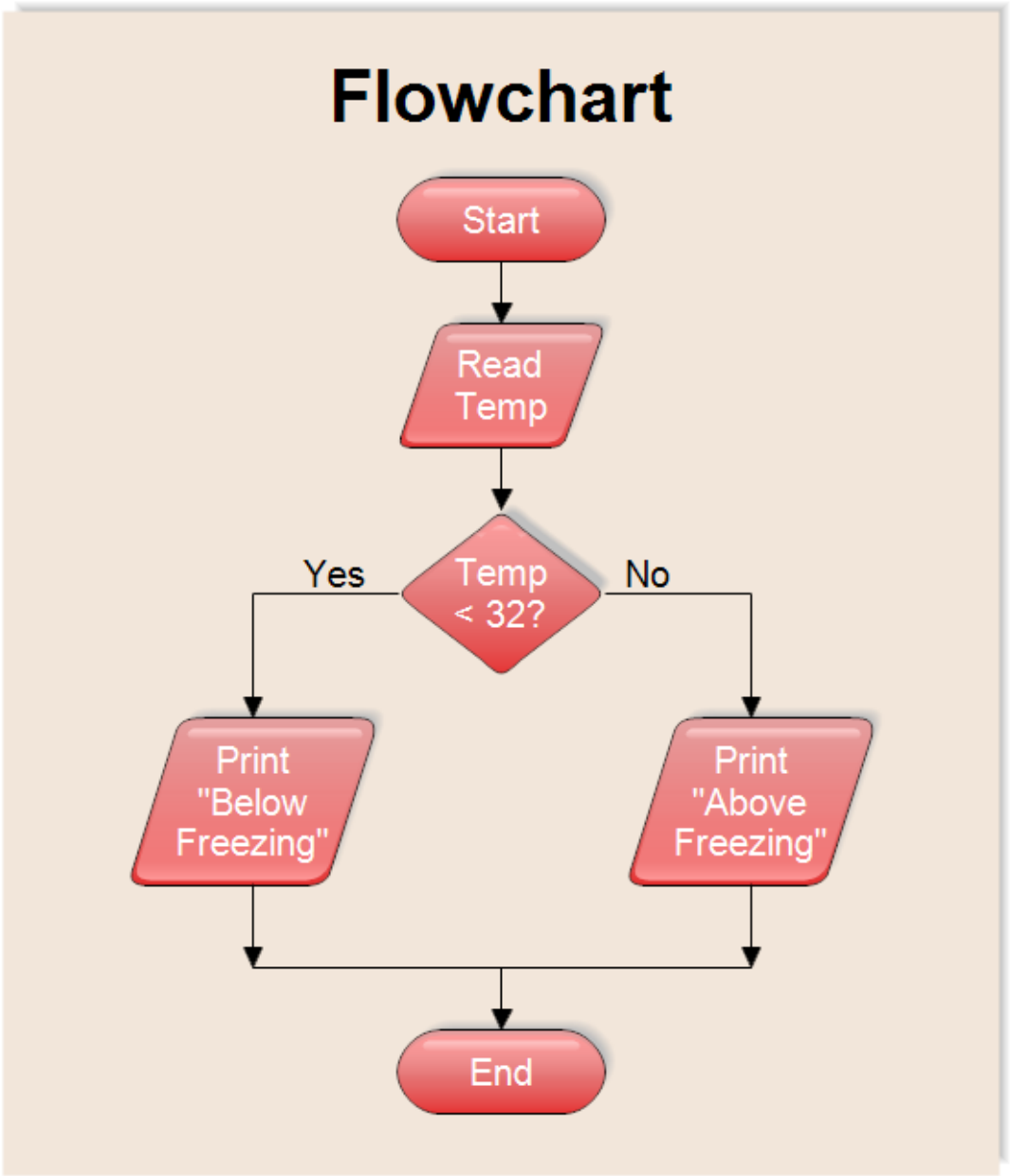
Input/ Output

Process



Condition

Flowchart: Example



Say Hello to Java!

The Java Platform consists of two parts:

- Java Virtual Machine (we've talked before),
- Java API (also called libraries).

Definition

The Application Programming Interface (API) is a huge collection of handy software packages that programmers can use.

Say Hello to Java!

The Java Platform consists of two parts:

- Java Virtual Machine (we've talked before),
- Java API (also called libraries).

Definition

The Application Programming Interface (API) is a huge collection of handy software packages that programmers can use.

You should install **Java SDK** (Software Development Kit) to bring Java programming platform into your machine. SDK includes programs such as:

- `java.exe`: executes Java applications
- `javac.exe`: Java compiler
- `javadoc.exe`: Javadoc generator

Programming Environment

There are two options to write, compile and execute your Java codes:

- Text editors + manual compilation and execution, also manual debugging.
  - **Visual Studio Code** (Cross-Platform)
- Integrated Development Environment (IDE); automatic compilation and execution + powerful debugging tools.
  - **IntelliJ IDEA** - Community edition (Cross-Platform)

Bugs!

Definition

- A **software bug** is an error, flaw, failure or fault in a computer program or system that causes it to produce an incorrect or unexpected result, or to behave in unintended ways.
- **Debugging** is the process of finding and resolving bugs or defects that prevent correct operation of computer software or a system.

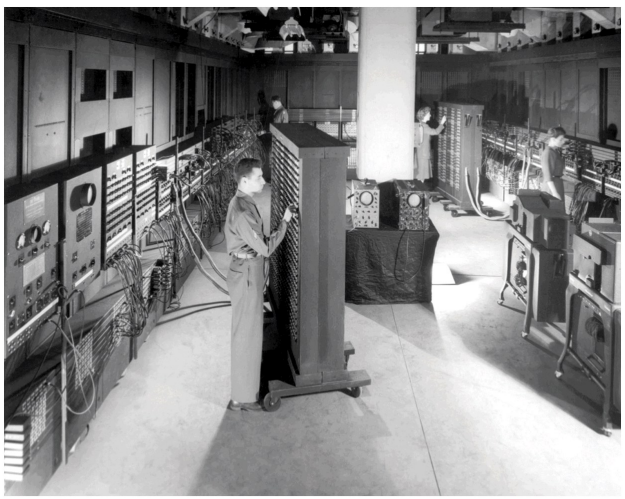


Figure 1.4
© UPPA/Photoshot

Your First Program

1. **Code**

   Open your favorite text editor and write the following code:

```
public class HelloPrinter
{
   public static void main(String[] args)
   {
      System.out.println("Hello, World!");
   }
}
```

   Save it as HelloPrinter.java

   Be careful:

   - Spelling is important.
   - JaVa iS CaSe SeNsItIvE.
   - Java uses special characters; e.g. { } ( ) [ ] ; "

Your First Program

2. **Compile**

   Open the console window, navigate to your code file,‘ and use **`javac.exe`** to compile your code according to the following format:

   `javac.exe code_file_name`

   Successful compilation creates a `.class` file.

   For example if we save the previous file in the root of drive ”C”, then we will write

   `javac.exe HelloPrinter.java`

Your First Program

3. **Run**

   Open the console window, navigate to the compiled file, and use **`java.exe`** to run your compiled code according to the following format:

   `java.exe class_name`

   For example in our previous compilation:

   ```
   java.exe HelloPrinter
   ```

Your First Program

3. **Run**

   Open the console window, navigate to the compiled file, and use **`java.exe`** to run your compiled code according to the following format:

   `java.exe class_name`

   For example in our previous compilation:

   ```
   java.exe HelloPrinter
   ```

   After hitting the enter key, you should see the following line:

   ```
   Hello, World!
   ```

From Source Code to Execution

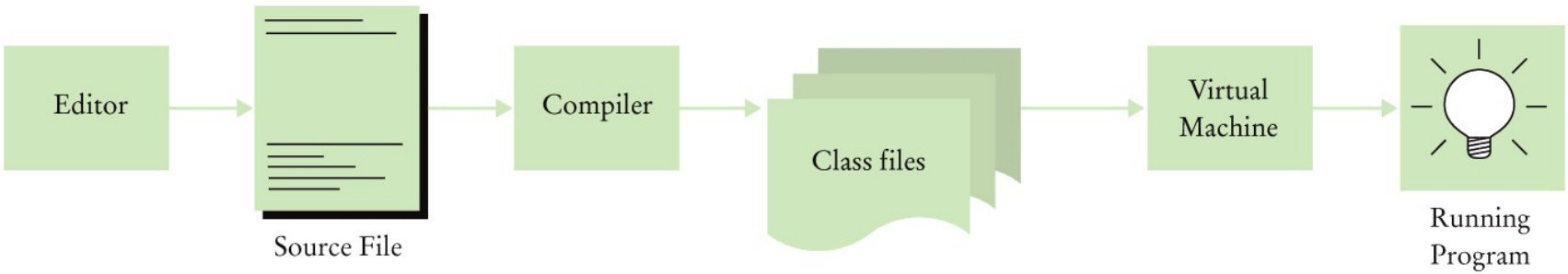
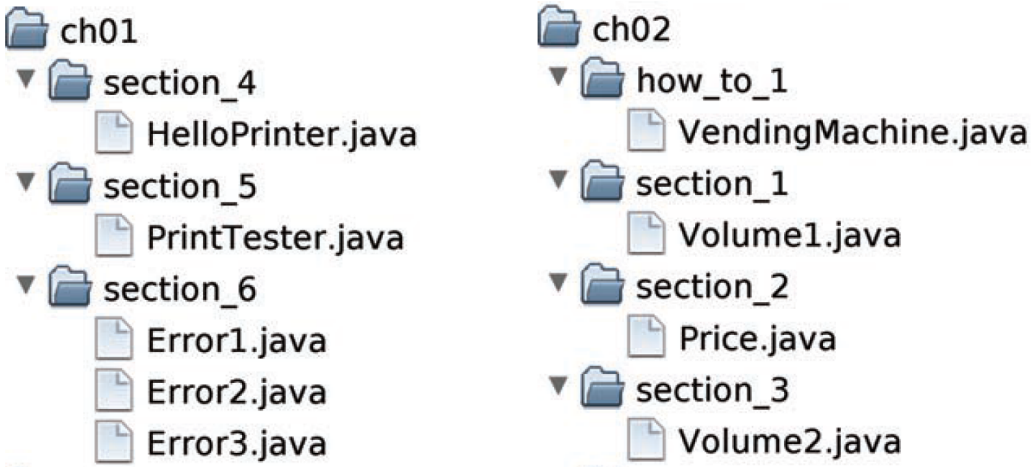


Figure 1.8

- The compiler generates the `.class` file which contains instructions for the Java Virtual machine.
- Class files contain 'byte code' that you **cannot** edit.

Organize Your Work



- Your 'source code' is stored in `.java` files.
- Create one folder per program
- **Backup your work!** (to a Flash Drive, external hard drive, or network drive)

Analyzing Your First Program

```
1  public class HelloPrinter
2  {
3     public static void main(String[] args)
4     {
5        System.out.println("Hello, World!");
6     }
7  }
```

1. Declares a '**`class`**' **`HelloPrinter`**.
   - Every Java program has one or more classes.
   - Classes are the fundamental building blocks of Java programs.
3. Declares a method called '**`main`**'.
   - Every Java application has exactly one '**`main`**' method.
   - The **entry point** where the program starts.
5. Method **`System.out.println`** outputs '`Hello, World!`'
   - A statement must end with a semicolon (`;`)

Minimal Java Program Template

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

Errors!

There are two categories of errors:

- **Compile-time Errors**
  - Syntax Errors
    Spelling, Capitalization, punctuation Ordering of statements, matching of braces/parenthesis.
  - No `.class` file is generated by the compiler.
  - Correct first error listed, then compile again.
- **Run-time Errors**
  - Logic errors
  - Program runs, but produces unintended results
  - Program may 'crash' (close or stop working suddenly).