CISC 3120

C02: Overview of Software Development and Java

Hui Chen

Department of Computer & Information Science

CUNY Brooklyn College

Outline

- Recap and issues
- Some concepts in software development
- Brief introduction to history of Java
- First Java program (from command line)
- Text-based application user interface
  - Command line arguments
- Review concepts of class and object
- Assignments

How well am I doing?

- Check "My Grades" CUNY Blackboard
  - Updated frequently

Attendance

- Y: you arrived on time and signed attendance sheet
- N: you missed the class
- L: you signed attendance sheet, but the instructor marked it as "late"
- X: you were excused with the support of legitimate documentation

Individual Assignment

- Practice W01-1_01-29
  - Individual assignment
  - A: Accepted
  - U: Unaccepted, not submitted, or not submitted on time.

Practice W01-1_01-29

- What shows up in Github classroom?
- Feedback method
  - Comments on commits
- Any problems?
  - Issue type 1: Accepted assignment, but did nothing. I don't even know who you are (unaccepted)
  - Issue type 2: The content of your file is wrong (unaccepted)
  - Issue type 3: You did not create the folder for the assignment (unaccepted)
- First assignment "give-away": fix your problem by before the class on Monday (February 5)

Group Assignment

- Project 0
  - Group assignment
- Letter grade
  - A: perfect or imperfect, I am very happy with what you team did.
  - B: at least three members completed the task.
  - C: at least two members completed the task.
  - D: at least one members completed the task.
  - F: nothing is done.
- Letter grade to be posted after the class, and only issued to contributors

Project 0

- What shows up in Github classroom?
- Feedback method
  - Comments on commits
- Any problems?
  - Issue type 1: some members did not join the team in Github
  - Issue type 2: some members joined, but did not make contribution
  - Issue type 3: contribution is being made, but content of the file does not meet requirement
- First team assignment "give-away": complete the task before the class on Monday, February 5.

Questions?

- Attendance
- Practice assignment
- Team Assignment
- Github issues

Suggestion

- Use Github Issue Tracking & Comments in your assignments
  - Practice assignment: manage your own to-do, bugs, and other items
  - Team project: manage to-do, bugs, other items; delegate tasks to members
- Communicate with the fellow team members and the instructor

Communicate Effectively

- Follow Stack Overflow,
  - https://stackoverflow.com/help/how-to-ask

Questions?

- What are you suggestions and comments?

Develop Software Applications

- Apply engineering principles and scientific techniques to software development
- Software engineering

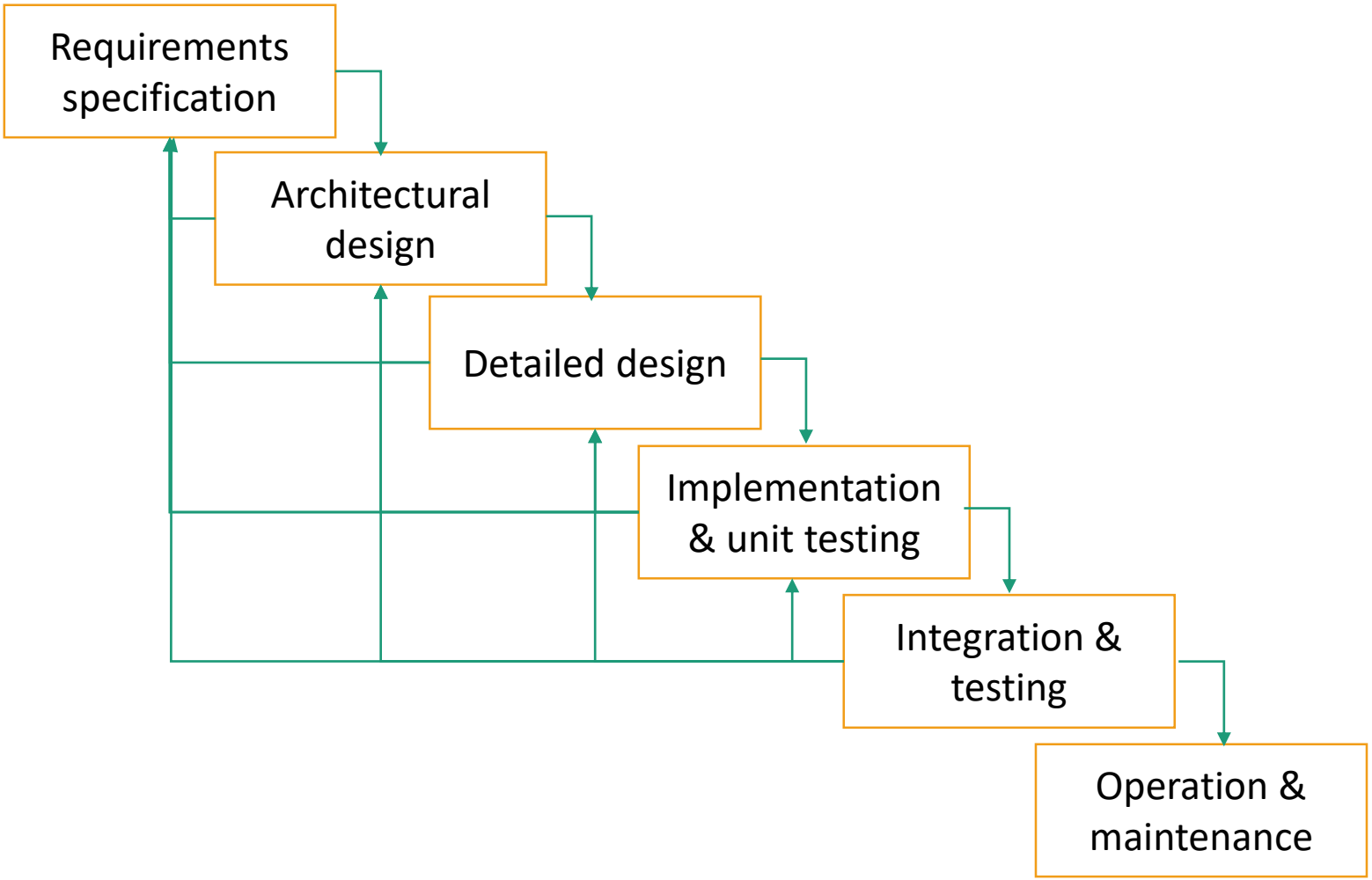
Main Parties in Software Development

- Two main parties: customers, designers
  - Real word scenarios can be more complicated about who the customers and designers are

Software Development Activities

- Main activities
  - Requirement analysis
  - Architectural specification
  - Detailed specification
  - Implementation and unit testing
  - Integration and testing
  - Operation and maintenance

Waterfall Model



Lots of Documentation

Software Development Process

- Waterfall vs. Spiral vs. Agile

Validation and Verification

- Validation
  - Are we building the right product?
- Verification
  - Are we building the product right?
- Testing

In CISC 3120 …

- Projects
  - The instructor is your customer
  - You and your team members are the designers
  - You deliver a verified product to the customer
- How does the software life cycle fit in?
  - Learn to design and develop applications in Java
    - Mostly concern about design, specification, implementation, and testing
    - Some object-oriented design and implementation

Java

- An Object-Oriented Programming language developed *after* C++
- Designed for devices, later for the Web
- "Write once, run anywhere"
- Java Virtual Machine (JVM)

Warm-up Group Discussion

- Read a Java program
  - Answer the given questions, and during the process, think about the following,
    - How much do you understand based on your knowledge and experience in C++?
    - What looks familiar to you?
    - What looks new to you?

Your First Java Program

- Set up environment
  - Check whether java and javac are present and what versions they are
  - Download and install JDK
- Create Java source code
  - Use a text editor (e.g., notepad)
- Compile Java source code to generate Java byte code
  - Use javac to compile the Java source code
- Run Java byte code in Java Virtual Machine
  - Use java to run the Java byte code program

The "Hello, World" Program

Create the HelloWorld.java file

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

Compile and Run

- From the command line:
  - Compile

    javac HelloWorld.java
  - Run

    java HelloWorld

Questions?

- Create Java program from scratch
- Compile Java program
- Run Java program

Command Line Interface

- Also called text-based interface
  - Method to interact with applications
  - Interaction is from text-based terminal
    - Text input only
- In Java, the interaction is done via the main method
  - public static void main(String[] args)
  - where "args" are the command line arguments

The Beer Song

- A little big program that prints out the lyrics of the song “99 bottles of beer”.

Interfacing BeerSong with Command Line Arguments

- What if I want to start at 10 beers?

Questions

- Using command line arguments
- Writing applications with text-based interface

Real-world Objects

- Examples
  - Chairs, desks …
  - Student, instructor …
- State and behavior

Software Objects

- State
  - States are stored in fields (member variables in C++, instance variables in Java, …)
- Behavior
  - Methods (member function in C++, instance methods in Java, …)
    - Methods operate on an object's internal state
    - Methods serve as the primary mechanism for object-to-object communication.

Data Encapsulation

- Hide internal state
- Require interaction to be performed through an object's methods

Using Objects

- Modularity
  - The source code for an object can be written and maintained independently.
- Information-hiding
  - With data encapsulation, the details of its internal implementation remain hidden from the outside world.
- Code re-use
  - Use objects written by you and other developers in many applications
  - Allow specialists to implement/test/debug complex, task-specific objects
- Pluggability and debugging ease
  - Can simply replace or diagnose problematic object

Real-world Classes

- Many individual objects are of the same kind
  - Many classes you can take
  - Many buses you can ride
  - There are many students
- They are of the same "blueprint"
  - A bus has different "color", "capacity", "route" …, i.e., has the same set of fields, but different values

Software Class

- A class is the blueprint from which individual objects are created
  - Fields and methods
- A class can have many instances

Design Java Program

- Identify objects and classes
- Identify fields and methods

The BeerSong Revisited

- More OOP way to write the BeerSong program
- Two objects
  - A BeerSong object
  - A BeerSongDriver object
  - Object-to-object interaction
    - The BeerSongDriver object calls the method of the BeerSong object

Questions

- Review a few concepts in Object-Oriented Programming
  - Objects
  - Classes
  - Fields & Methods
  - State and Behavior

Assignments

- Assignment posted the Class Website and the Blackboard
  - Practice assignment
  - CodeLab assignment