

ADVENTIST UNIVERSITY OF CENTRAL AFRICA

DEPARTMENT OF SOFTWARE ENGINEERING

Course: Programming best practices and design patterns

IVAN K. Bisengimana

OBJECTIVES OF THE COURSE

The objective of this course are as follows:

- To teaches the student how to apply programming best practices
- To Teach student how to avoid pitfalls.
- Student will be able to understand process of software development life cycle.
- To teach and inform perform industry-standard software development techniques.
- To explain design patterns to implement general reusable solutions to reoccurring problems.
- Student will be able to apply common design pattern.
- student will be able to software development concepts

GENERAL INTRODUCTION

What is programming?

- Programming refers to a technological process for telling a computer which tasks to perform in order to solve problems.
- You can think of programming as a collaboration between humans and computers, in which humans create instructions for a computer to follow (code) in a language computers can understand.

CONT(...)

- Programming enables so many things in our lives. Here are some examples:
- When you browse a website to find information, contact a service provider, or make a purchase, programming allows you to interact with the site's on-page elements, such as sign-up or purchase buttons, contact forms, and drop-down menus.

CONT(...)

- The programming behind a mobile app can make it possible for you to order food, book a rideshare service, track your fitness, access media, and more with ease.
- Programming helps businesses operate more efficiently through different software for file storage and automation and video conferencing tools to connect people globally, among other things.
- Space exploration is made possible through programming.

HOW DOES COMPUTER PROGRAMMING WORK?

- At its most basic, programming tells a computer what to do. First, a programmer writes code—a set of letters, numbers, and other characters.
- Next, a compiler converts each line of code into a language a computer can understand. Then, the computer scans the code and executes it, thereby performing a task or series of tasks.
- Tasks might include displaying an image on a webpage or changing the font of a section of text.

TYPES OF PROGRAMMING LANGUAGES

- While hundreds of programming languages (if not thousands) exist, and there are many ways to classify them, they typically can be grouped into five main categories:
- Procedural programming languages: is the standard approach used in traditional *computer language* such as C, Pascal, FORTRAN & BASIC
- Functional programming languages: are a class of languages designed to reflect the way people think mathematically, rather than reflecting the underlying machine.

CONT(...)

- Object-oriented programming languages: Languages that support the procedural programming paradigm are: C, C++, Java, Pascal and etc
- Scripting languages: Scripting is the action of writing scripts using a scripting language, distinguishing neatly between programs, which are written in conventional programming language such as C,C++,java, and scripts, which are written using a different kind of language.
- Logic programming languages: which is largely based on formal logic.Any program written in a logic programming language is a set of sentences in logical form, expressing facts and rules about some problem domain

HOW TO GET STARTED IN COMPUTER PROGRAMMING

- Identify your programming goals.
- Enroll in an introductory programming course.
- Build essential programming skills.
- Connect with other programmers.
- Complete projects.
- Explore your future in programming.

WHAT IS A SOFTWARE

- As you know computer cannot do anything on its own and has to be guided by the user.
- In order to do any specific job you have to give a sequence of instructions to the computer.
- Software refers to the set of computer programs, procedures that describe the programs and how they are to be used.
- We can say that it is the collection of programs, which increases the capabilities of the hardware.
- Software guides the computer at every step where to start and stop during a particular job. The process of software development is called programming

CONT...

- Types of software are:
 - ❑ application software
 - ❑ System software programs
- System software programs work on its own while application software is dependent on it.
- Application software is mainly classified into category
 - Window application and
 - Web application

PROGRAMMING BEST PRACTICE

- Writing good software is not an easy task, it requires a lot of coding experience and skills.
- Therefore, inexperienced software developers or newbies suffer from this critical task.
- In this course, we will provide guidelines to help in this important context.

CONT(...)

- It presents the most important best practices and recommendations of writing good software from software engineering perspective regardless of the software domain (whether for desktop, mobile, web, or embedded), software size, and software complexity.
- The best practices that will be provided are organized in taxonomy of many categories to ease the process of considering them while developing software.
- Furthermore, many useful, practical, and actionable recommendations are given mostly in each category to be considered by software developers.

-
- writing software is the art and process of converting what is in mind into reality.
 - This process aims to produce software-based solutions for real-world problems that we encounter in our everyday life.
 - Software is everywhere around us and it changes the way people live.
 - As the software engineer Mark Andreessen (the co-founder of Netscape, one of the first web browser companies) said,
“Software is eating the world”.

CONT(...)

- More and more stuff are getting automated to make our life easier, faster, and smarter, and as much more automation is going to be just a function of software.
Software plays a dual role. It is a product and a vehicle for delivering products at the same time.
- As a product, it is created to deliver its functionality to users via various forms including software applications, web applications, web services, embedded systems, etc.

CONT(...)

-
- As a vehicle for delivering products, software constitutes the foundation for creating, managing, controlling, etc. of other software products such as software tools, frameworks, and platforms

BEST PRACTICES AND RECOMMENDATIONS

- In this section, the proposed taxonomy that assists developers to create good software illustrated in categories.
- The categories cover different aspects of writing any software and they are based on software engineering, industry, and coding experiences

CONT (...)

- The best practices and recommendations that have to be taken into consideration are as follow:
 - ❑ Before Writing Software
 - ❑ While writing software and
 - ❑ After Writing a Software

BEFORE WRITING SOFTWARE

- Build a good team
- Work with professionals
- Understand software categories
- Select a development model.
- Select an advanced IDE
- Select a user interface
- Use programming language features
- Select a programming language
- Select software architecture

WHILE WRITING SOFTWARE

- Avoid quick coding
- Follow coding standards
- Simplify coding
- Reuse existing code
- Minimize code dependences
- Focus while coding
- Follow software evolution
- Use comments
- Avoid code duplication
- Use programming concept

AFTER WRITING SOFTWARE

- Review code
- Test code
- Maintain code
- Create a user manual
- Make searchable
code

WHAT IS PITFALLS IN PROGRAMMING

- It is crucial for educators to understand pitfalls for novice programmers in programming education at computer science/Engineering course.
- By giving well-timed advice to students who fall into a pitfall, educators are able to achieve efficient teaching and keep up their students' motivation.
- However, so far, it is less well-understood how novice students edit source code and why they fall into a pitfall.

CONT...

- As computer technology offers efficient and high performance information processing, it has got popularity over the home and office users in the whole world.
- Software development methodologies are management practices for software development projects.

CONT(...)

- we're going to present some of the most common mistakes which every Java programmer eventually makes, the reasons why they happen and solutions to avoid them.
- Note that making mistakes are not bad. It is good for helping programmers realize and understand the problems better as well as improve their programming skills.
- It's important to learn lessons from mistakes, and the more mistakes you made, the more lessons you can learn, the more experience (and wisdom) you have.

CONT...

- 1. Accessing non-static members from static methods (mostly from the main method)
- 2. Missing closing curly braces
- 3. Missing break in switch case construct
- 4. Confusing assignment with comparison (= and ==)
- 5. Confusing object comparison (== instead of .equals)
- 6. Confusing about 0-based or 1-based index
- 7. Using less restrictive access modifiers
- 8. Forgetting to free resources
- 9. Ignoring Exceptions
- 10. Modifying a collection while iterating it

CONT...

- **Accessing non-static members from static methods (mostly from the main method)**

```
public class StaticExample1 {  
    public int number; // instance variable  
  
    public static void main(String[] args) {  
        number = 30;    // compile error  
    }  
}
```

CONT...

Missing closing curly braces

refer to the previous example

```
public class StaticExample1 {  
    public int number; // instance variable  
  
    public static void main(String[] args) {  
        number = 30;    // compile error  
    }  
}
```

CONT...

- **Missing break in switch case construct**

```
Scanner scanner = new Scanner(System.in);
int option = scanner.nextInt();

switch (option) {
    case 1:
        System.out.println("1 -> New");
        break;
    case 2:
        System.out.println("2 -> Edit");
        break;
    case 3:
        System.out.println("3 -> Delete");
        break;
    case 4:
        System.out.println("4 -> View");
        break;
    case 5:
        System.out.println("5 -> Quit");
    default:
        System.out.println("Unknown command");
}
```


CONT...

- **Confusing assignment with comparison (= and ==)**

The = is an assignment operator whereas the == is a relational operator, and sometimes one mistakenly uses = instead of ==. Take a look at this code snippet:

Confusing object comparison (== instead of .equals)

The == operator compares two objects physically (their memory addresses) whereas the equals() method compares two objects semantically (their information). And in most of the cases, we should compare two objects meaningfully using the equals() method.

- **Confusing about 0-based or 1-based index**

Like many other programming languages, array index in Java are 0-based, which means the first element in the array starts at the index 0, the second at 1, and so on. However, sometimes we mistakenly treat the first element at index 1.

```
1 | String[] fruits = {"Apple", "Banana", "Carrot", "Grape"};  
2 | String firstFruit = fruits[1]; // should be: fruits[0]
```

CONT...

- **Using less restrictive access modifiers**
- This is another very common mistake of all beginners: using access modifiers without taking care of its visibility. The Java language provides 4 access modifiers for protecting members of a class:

public > protected > default > private

These access modifiers are sorted from the least restrictive (**public**) to the most restrictive (**private**). However, novice programmers do not fully understand this and often use the **public** and default access modifiers which make the code less secured.

SUMMARY OF ACCESS MODIFIER

	Private	Protected	Public	Default
Same class	YES	YES	YES	YES
Sub package subclass	NO	YES	YES	YES
Same package non-subclass	NO	YES	YES	YES
Different package subclass	NO	YES	YES	NO
Different package non sub-class	NO	NO	YES	NO

CONT...

- **Forgetting to free resources**

This is also a common mistake which one easily forgets to close resources after use, such as network connections, database connections, file streams, etc.

This mistake may lead to resources leaked or memory occupied by no longer used objects.

Ignoring Exceptions

This is another common mistake of beginners who are too lazy to write code for handling exception. It seems to be of no harms if the code runs fine without exceptions. However, in case the exceptions occurred, the code can fail silently which adds difficulty in finding the problem.

CONT...

- **Modifying a collection while iterating it**

Anyone who starts programming with Java makes this common mistake: attempt to modify (remove) elements in a collection while iterating over it.

SYSTEM DEVELOPMENT LIFE CYCLE (SDLC)

- The System Development Life Cycle (SDLC) is a conceptual model for software development that divides up the process into different phases. Each phase has a distinct role to play in the development life cycle, and is a building block for the next phase.

There are many SDLC models, each emphasizing different aspects of the life-cycle.

Implementing a structured approach requires selecting a model and utilizing it throughout the development life cycle



CONT...

- The following seven SDLC phases illustrate the selected Waterfall Model
 1. Software Concept
 2. Analysis
 3. Design
 4. Coding and Debugging
 5. System Integration and Testing
 6. Implementation
 7. Maintenance and Support

SOFTWARE DEVELOPMENT METHODOLOGIES

- Types of software development methodology:
 - ❑ waterfall
 - ❑ Agile
- The key difference between Agile vs. Waterfall is that Waterfall breaks down software development into isolated phases that flow into each other, while Agile advocates iterative development cycles in which multiple lifecycle phases can run in parallel.
- Whether an organization develops applications with an Agile- or Waterfall-based methodology, all software development projects incorporate common aspects of lifecycle phases, including:

CONT(...)

- requirements gathering;
- analysis and design;
- development and implementation;
- testing and verification; and
- deployment and ongoing maintenance

CONT(...)

How does **Waterfall** work?

- The Waterfall model treats each software development lifecycle phase as distinct, isolated events. Development teams must complete each phase in its entirety before they move onto the next phase.
- Furthermore, once teams move onto a new phase, they are not allowed to revisit a previous one.
- Progress through the development phases flows in only one direction, from the top to the bottom -- just like flow of a waterfall, which is where the model's name comes from.
- Assignment: **Why does Agile reject Waterfall?**

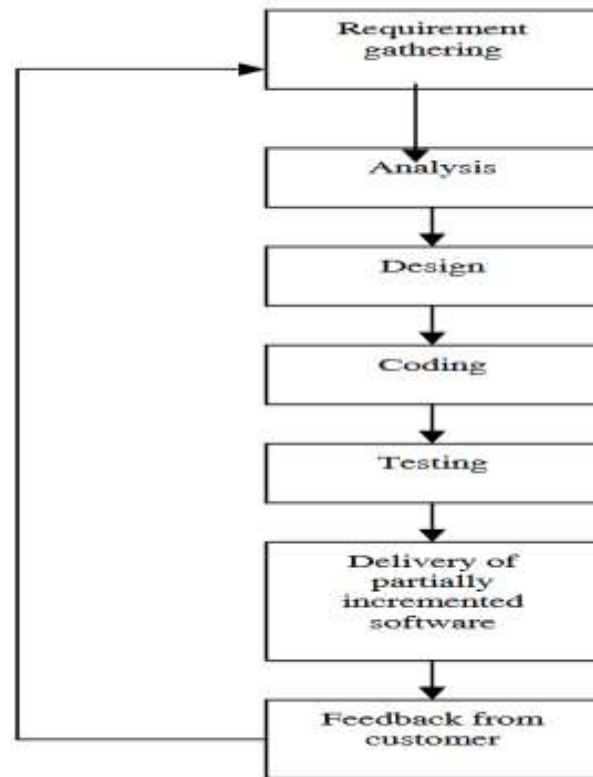
AGILE METHOD

- In software development life cycle, there are two main considerations, one is to emphasize on process and the other is the quality of the software and process itself.
- Agile software processes is an iterative and incremental based development, where requirements are changeable according to customer needs.
- It helps in adaptive planning, iterative development and time boxing. It is a theoretical framework that promotes foreseen interactions throughout the development cycle.

CONT...

- Agile process is an iterative approach in which customer satisfaction is at highest priority as the customer has direct involvement in evaluating the software.
- The agile process follows the software development life cycle which includes requirements gathering, analysis, design , coding , testing and delivers partially implemented software and waits for the customer feedback.
- In the whole process , customer satisfaction is at highest priority with faster development time.

AGILE PROCESS DIAGRAM



CONT...

-
- Lets explain this using **AUCA Registration System**

Lets assume in this application we have 4 module:

- ☐ **Registration module**
- ☐ **Course assignments module**
- ☐ **Schools fees payments module**
- ☐ **Mark sheet module**

CHARACTERISTICS OF A GILE P ROJECTS

- Agile process requires less planning and it divides the tasks into small increments. Agile process is meant for short term projects with an effort of team work that follows the software development life cycle.
- Software development life cycle includes the following phases
 - ❑ 1.Requirements gathering,
 - ❑ 2.Analysis,
 - ❑ 3.Design,
 - ❑ 4.Coding
 - ❑ 5.Testing,
 - ❑ 6.Maintenance.

CONT...

- The involvement of software team management with customers reduces the risks associated with the software.
- This agile process is an iterative process in which changes can be made according to the customer satisfaction.
- In agile process new features can be added easily by using multiple iterations

KEYS ASPECT OF AGILE METHODOLOGY

- **1. Iterative**

The main objective of agile software processes is satisfaction of customers, so it focuses on single requirement with multiple iterations.

- **2. Modularity**

Agile process decomposes the complete system into manageable pieces called modules. Modularity plays a major role in software development processes

- **3. Time Boxing**

As agile process is iterative in nature, it requires the time limits on each module with respective cycle.

CONT...

4. Parsimony

In agile processes parsimony is required to mitigate risks and achieve the goals by minimal number of modules.

5. Incremental

As the agile process is iterative in nature, it requires the system to be developed in increments, each increment is independent of others, and at last all increments are integrated into complete system.

CONT...

6. Adaptive

Due to the iterative nature of agile process new risks may occurs. The adaptive characteristic of agile process allows adapting the processes to attack the new risks and allows changes in the real time requirements.

7. Convergent

All the risks associated with each increment are convergent in agile process by using iterative and incremental approach.

CONT ...

9. Collaborative

As agile process is modular in nature, it needs a good communication among software development team.

Different modules need to be integrated at the end of the software development process.

10. People Oriented

In the agile processes customer satisfaction is the first priority over the technology and process.

A good software development team increases the performance and productivity of the software.