




OOP

Object Oriented Programming

for: SCII- SCCI- & SCCJ- 2021
Jan - April 2024



By: Salesio M. Kiura
Department of Computer Science and Informatics
School of Computing and Informatics,
Technical University of Kenya (TU-K)



3



Session 3:

Review of OOP Analysis as pre-cursor to development, Lab Session

.....Smoooooth flow to java

Date: __6th February 2024__Face2Face__

Recap

Java set up (JDK/JRE), compilation and execution commands

Keywords:

- ▀ java files - .java, **public classes** are in files with the **same name** as the class name,
- ▀ object → class → object,
- ▀ class (definition: properties 'variables' and behaviors methods),
- ▀ **main method** special - JVM, main method signature – **static public void main(Strings argv[])**,
- ▀ java compiler, bytecode 'magic of java', Java Virtual Machine, java Launcher
 - ▀ Java is machine independent – compile once execute many times

Recap

So far ...

- ▀ Course Objectives, Learning outcomes
- ▀ Paradigms and languages, comparison and contrast
- ▀ Merits and Demerits of OOP
- ▀ Illustrative examples
- ▀ Basic OOP concepts
- ▀ OOP Analysis: Case studies

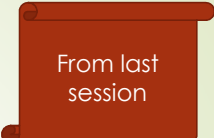


■ Now the Assignment



EXAMPLE

- 1. Part of a Police System ?
- 2. Part of a Health Center System
- 3. TUK innovations
 - Person Identification system (Part of security arrangements initiative)
- 4. Automation of a land registry
 - Ref. "Land Registration (General) Regulations 2017" – pages 6 and 7
 - See Part II – of the regulations titled:
Organization and Administration of Registries
 - we go through the overview of what happens at the registry
pages 6 and 7



From last
session

- In the context of the ongoing police reforms in Kenya, the Officer Commanding (a local police) Station has approached you to design a simple system for the station. The idea is to automate/computerize the **police Occurrence Book (OB)**. The OB is used to record complaints from the wananchi at the police station. A record of a complaint shows, among other things the following:

From last session

- The police officer who attended to the mwananchi, the date of incidence, details of the person reporting, the report from the mwananchi, etc
- After listening to the mwananchi, the police officer must indicate on the form whether the reported incident is a theft, murder, disagreement, accident or any other

1

- You are part of a team of developers mandated to come up with a **hospital information management system**. The system will cover the entire processes of attending to patients. From an initial meeting with the head of the medical records at the hospital, you have found out the following.

- ... the hospital has a casualty department that acts as the first point of contact with patients. Once a patient arrives at the casualty, his/her details are taken starting with his names. Given the names, the clerk is able to search the records and find out if this is a returning patient or a first time client of the hospital. If the patient is a returning one, he/she pays a registration fee of Kshs. 50 and proceeds to the triage where his/her vital signs (blood pressure, pulse rate, height, age, temperature, etc) are recorded before proceeding to the consultations department to see a clinician who is able to diagnose the ailment of the patient. A first time patient is required to provide more details such as: date of birth, address (where resident), next of kin, etc. he/she pays 100 registration fees and proceeds to the triage from where hr/she is attended to just like a continuing patient.

From last session

....

2

- In the context of the ongoing modernization of TU-K, the University head of Security has approached you to design a simple system for the **askaris** manning the University gate. The idea is to automate / computerize what happens at the gate. Starting with the Visitors-Book. The visitors book records all visitors to the University premises. At any given gate, a record of a visitor shows, among other things the following:

From last session

- The officer who attended to the visitor, the date of visit, details of the visitor, destination point/office, the purpose/objective of the visit, mode of traveling used (and corresponding details), report from the visited person, gate used to exit, etc
- After listening to the visitor, the security officer (**askari**) must indicate on the form whether the purpose of visit is official, private, or returning resident. Official can be administrative office visit, lecturing, studying, working,
- etc

3

Exercise

From last session

- Required:
 - Identify objects
 - identify **relevant** data (members of the class)
 - Identify **relevant** methods (behaviors and messages)

Object Oriented Programming in practice

- We said Object Oriented programming is about:
 - objects and assigning responsibilities
 - Objects communicate to other objects by sending messages
 - Messages are received by the methods of an object

Exercise Task Description

- The exercise is done in Groups of at most 5 students and minimum of 3 students.
- Formation of groups (random allocation of a group by the web conferencing system)
- NB
 - Once in the groups (Break out rooms for 20 Mins):
 - Elect a leader who will coordinate preparation of the document to be submitted
 - Agree on what case to work on and make a copy of the respective file. Name the file as Group_XX_Case_y - tttt.docx where: xx is the group number, y is either 1 or 2 or 3 and tttt is the title of the case. Ideally only group number is updated. Example **Group_03_Case 3 – Police reforms system.docx**
 - The lecturer will share the files in the eLearning platform
 - An invitation to a shared folder will be shared where each group should save the file they are working on.

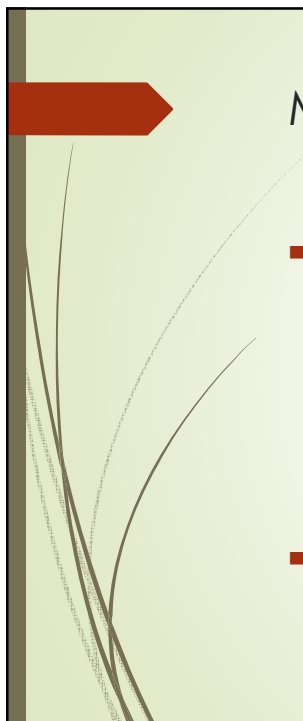
From last session

➤ Your Identified Objects


➤ ...

➤ ➔ Online shared word documents

Every Group leader who submitted got a link to document with annotations and comments

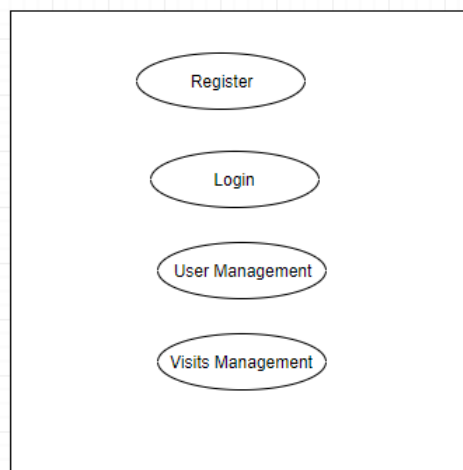
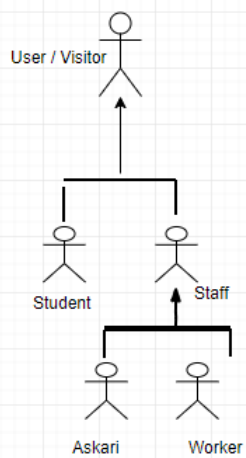
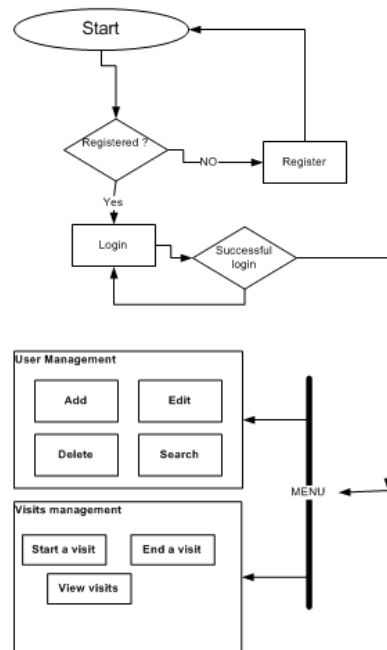


Mini projects



- With the above description as an example, you are required to **identify and describe a scenario** that can form the basis of a semester mini-project
 - Deliverable: A flow chart (don't mind the completeness/correctness at this time)
 - An initial "general overview" class diagram representing the project's case
- Due in next class!
 - Group work

Example of deliverable





Lab Class Session

DATE: __6th Feb 2024__

Venue: __D13__



Introduction - Java Development Technologies

- Java Development Kit (JDK)
 - a package of tools for *developing* Java-based software
- Java Runtime Environment
 - the on-disk part of Java that creates the JVM. The JRE contains tools for *running* Java code
- Java Virtual Machine
 - The Java platform component that executes programs

The JRE can be used as a standalone component to simply run Java programs, but it's also part of the JDK. The JDK requires a JRE because running Java programs is part of developing them.

Getting started with java (1 of 2)

GraalVM

- GraalVM – a high-performance JDK distribution. It is designed to accelerate the execution of applications written in Java and other JVM languages
- Getting GraalVM installed and ready-to-go should take a few minutes
 - <https://www.graalvm.org/>
 - Downloads (**GraalVM Community** – a Community supported **open source** build)

Download Link

- Direct download Link:

■ <https://github.com/graalvm/graalvm-ce-builds/releases/tag/vm-22.1.0>

Platform	Java 11	Java 17	
Linux (amd64)	 download	 download	instructions
Linux (aarch64)	 download	 download	instructions
macOS (amd64) +	 download	 download	instructions
macOS (aarch64/M1) +	 download	 download	instructions
Windows (amd64)	 download	 download	instructions

- This distribution of GraalVM Community includes:

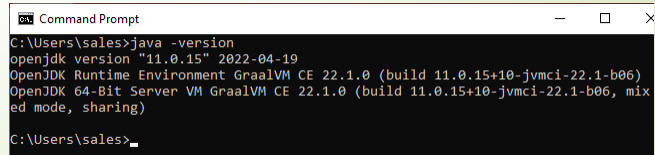
- Java runtime with the GraalVM compiler
- JavaScript runtime
- LLVM runtime
- Monitoring and debugging tools

- The GraalVM environment can be extended with optionally available components for web development environments and frameworks



Installation guides

- Windows:
 - [Installation on Windows Platforms \(graalvm.org\)](https://www.graalvm.org/22.1/docs/getting-started/windows/)
 - <https://www.graalvm.org/22.1/docs/getting-started/windows/>
- Linux platform
 - [Installation on Linux Platforms \(graalvm.org\)](https://www.graalvm.org/22.1/docs/getting-started/linux/)
 - <https://www.graalvm.org/22.1/docs/getting-started/linux/>
- Others:
 - <https://www.graalvm.org/22.1/docs/getting-started/>
- Test the installation of JDK
 - `java -version`



```

C:\Users\sales>java -version
openjdk version "11.0.15" 2022-04-19
OpenJDK Runtime Environment GraalVM CE 22.1.0 (build 11.0.15+10-jvmci-22.1-b06)
OpenJDK 64-Bit Server VM GraalVM CE 22.1.0 (build 11.0.15+10-jvmci-22.1-b06, mixed mode, sharing)

C:\Users\sales>_
  
```

Text based HelloWorld App execution

- In VS code
 - With a text editor:

```

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

- `C:\your_Folder> javac HelloWorld.java`
- `C:\your_Folder> java HelloWorld`
Hello World Class!

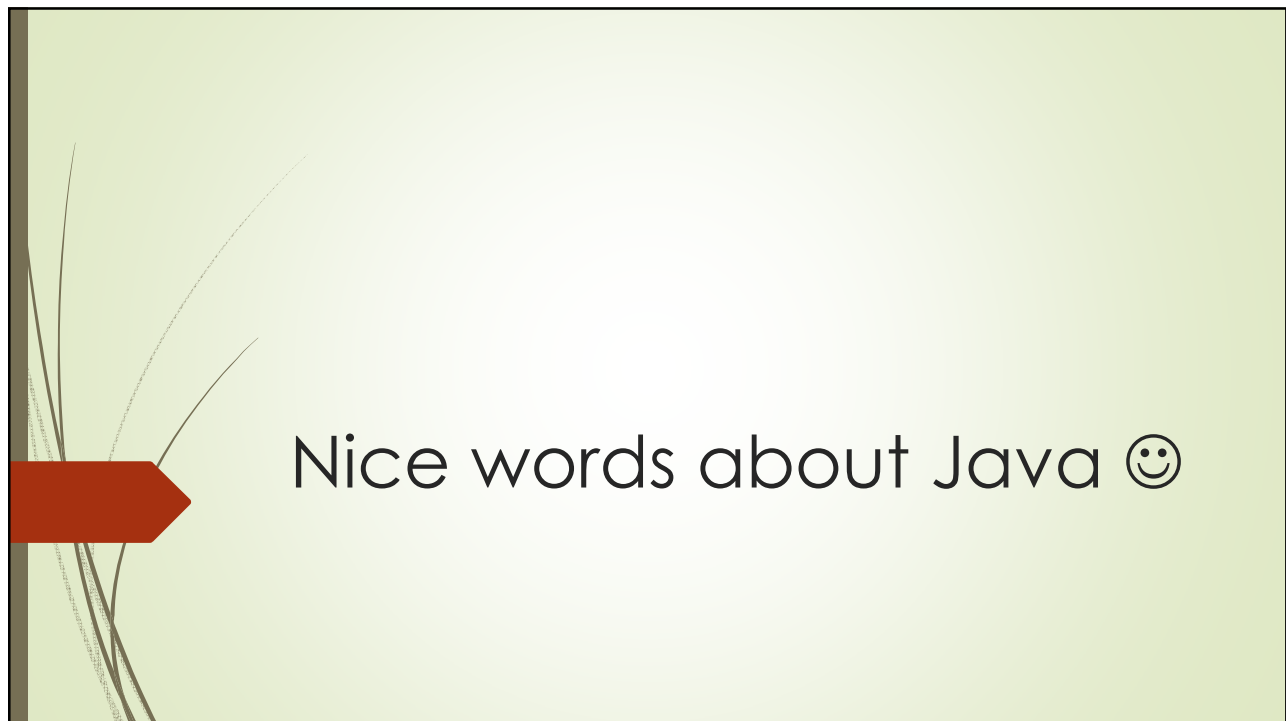
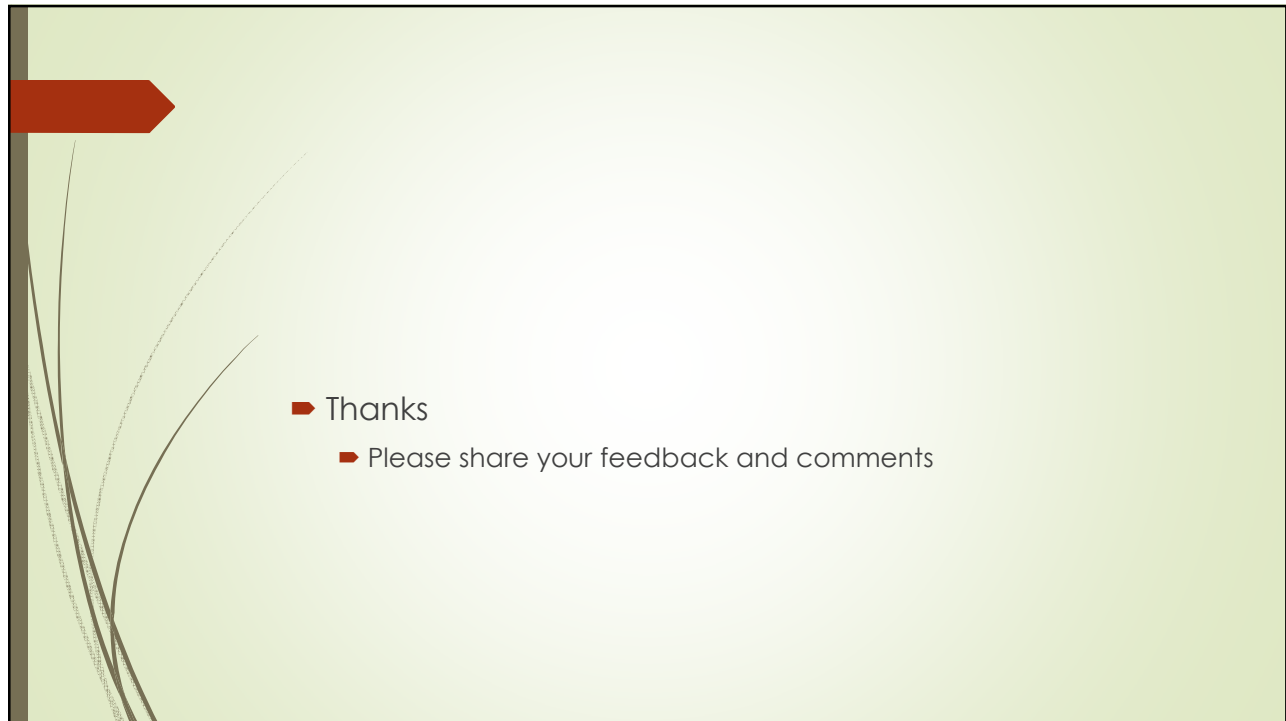
Getting started with java NetBeans or any other IDE

The motivation would be to avoid having to download a single installation file that sets up the necessary development environment

- Apache NetBeans (<https://netbeans.apache.org/>)
 - Fits the Pieces Together,
 - **Development Environment**, Tooling Platform and Application Framework.
 - More than an editor!
 - Download links (choose a version of your choice)
 - <https://netbeans.apache.org/download/>
 - Installation is a standard process
 - The system should recognize the earlier jdk folder (e.g. GraalVM)

HelloWorld Code

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



About JAVA

- Is used for creating:
 - intelligent consumer-electronic devices (cell phones)
 - Web pages with dynamic content
 - large-scale enterprise applications

Java life cycle

- Java programs normally undergo four phases:
 - **Edit** (Source code (.java))
Programmer writes program (and stores program on disk)
 - **Compile** (Byte codes (.class) , as (.exe) in c++)
Compiler creates bytecodes from program (.class as .exe in c++)
 - **Load**
Class loader stores bytecodes in memory
 - **Execute**
Interpreter: translates bytecodes into machine language



■ Other concepts

- The Java Application Programming Interface (API)
 - a large collection of ready-made software components. It is grouped into libraries of related classes and interfaces; these libraries are known as packages.
 - E.g. `System.out.*`; `java.util.*`
- Java Virtual Machine (JVM)
- Machine code (platform dependent)



Simple Exercise

- All participants to “AT LEAST” be able to code:
- Hello WORLD in JAVA!
 - Using an IDE of your choice
 - Hand coded Hello World
 - On command prompt
 - On a GUI (message Dialog Box)
 - Interactive Hello World
 - Enter on command prompt
 - Enter using an input dialog box

