

Programming Fundamentals (PRF)

◆ What is a Programming?

Programming refers to a technological process for telling a computer which tasks to perform in order to solve problems.



Gatalu wisadeema sadaha bawitha karana technical process ekak

- Programming is a technical process used to solve real world problems

◆ What is a Programming Language

A programming language is a set of instructions and syntax used to create software programs.



Software program ekak liyanna baawitha karana language ekak.

- set of instructions and syntax used to create software programs.

(software program ekak liyanna instruction saha syntax set ekak use karala liyana language ekak)

◆ Generations of programming Languages

- 1st Generation Languages
- 2nd Generation Languages
- 3rd Generation Languages
- 4th Generation Languages
- 5th Generation Languages

Low Level Languages

High Level Languages

● 1st Generation Languages(1GL)

1st Generation Language, also known as the Machine language, is the most basic programming language that computers can comprehend. It operates at the machine level and consists of a series of binary codes represented by 0s and 1s.

```
0001111100001010101  
0011100111001101010  
0101010101010000000  
1010101010101010101  
1010100000111100000  
1010101000111000101  
1010101010010100100
```

E.g. = Machine language

1GL ----- (Binary code) -----> Device

(Computer එකක් වැඩ කරන්නේ current එක ලැබීම සහ නොලැබීම මත)

● 2nd Generation Languages(2GL)

Second-generation languages, known as assembly languages, introduced human-readable notations, such as mnemonics, that are easier to understand than binary code. These languages still require conversion to machine code through an assembler before execution.



E.g. = Assembly

2GL -----> -----> -----> Device

(Human readable language Note)

Instruction දෙන වැශේ 1GL ට වඩා easy

(Compiler)

● 3rd Generation Languages(3GL)

Third-generation languages, also known as procedural languages, are high-level programming languages designed to be more user-friendly by utilizing syntax similar to human language. This makes it simpler for programmers to create and understand code.



E.g = C, C++, C##, Java, python



● 4th Generation Languages(4GL)

Fourth-generation languages (4GL) are created to simplify programming by using syntax that is more similar to everyday human language. These languages allow users to focus on specifying what tasks need to be done, without needing to worry about the specific implementation details.



E.g. = R languages



● 5th Generation Languages(5GL)

Fifth-generation languages (5GL) represent the latest stage in programming evolution and focus on visual programming and artificial intelligence. These languages use visual tools and constraint-based logic to create programs. Instead of writing step-by-step instructions, the programmer defines goals, and the system generates the code to achieve them.



E.g= OPS5, Mercury



Me anuwa generation eka aluth wenna aluth wenna instructions dena wade lehesi welaa thiyanawa

● What is Software ?

අපට document එකක් edit කරන්න ඕන නම් අපි Microsoft word වැනි දෙයක් use කරනව , අන්න ඒ process එක Software එකක්.

- Software is a set of instructions that tells a computer what to do.

Java



◆ **What is Java?**

Java is a high-level, general-purpose, memory-safe, object-oriented programming language.

◆ **Reason for Java**

- high-level (instruction දෙන්න easy),
- general-purpose(පොඩි solution එකක ඉදත් advance solution වෙනකල් generally use කරන්න පුළුවන්),
- memory-safe, (Java memory safe වෙන්නේ pointer arithmetic නැති එක, array bounds check තියෙන එක, type check කරන එක, සහ Garbage Collector එක මතකය auto manage කරන එක නිසා.)
- object-oriented (java කියන language එක හැම දෙයක්ම සලකන්නේ object එකක් විදිහට නිසා)

◆ History Of Java

- Java was designed and developed at **Sun Microsystems**.
- Java language is developed by **James Gosling** in the **early 1990s**.
(1995 - Officially released Java 1.0) The team initiated this project to develop a language for digital devices such as set-top boxes, television, etc.



- Originally C++ was considered to be used in the project but the idea was rejected for several reasons(For instance C++ required more memory).
{C++ වල memory එක ගොඩක් ගන්න case එකක් තිබේ නියෙනව }
- Gosling endeavoured to alter and expand C++ however before long surrendered that for making another stage called Green.
- James Gosling and his team called their project "Greentalk" and its file extension was .gt
- GreenTalk later became to known as "**OAK**".
- The name Oak was used by Gosling after an oak tree that remained outside his office.



- But they had to **later rename it as "JAVA" in 1995** as it was already a trademark by Oak Technologies.
- The name Java originates from a sort of espresso bean, Java. Gosling came up with this name while having a coffee near his office.



Summary

- Java was designed and developed - James Gosling (1995 - Java 1.0).
- Developed company- Sun Microsystem
- Greentalk → OAK → JAVA(1995)

What is the usage of Java?

- Mobile applications (specially Android apps)
- Desktop applications
- Web applications
- Web servers and application servers
- Games
- Database connection and etc.



Why use Java?

- Java works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.)
- It is one of the most popular programming languages in the world
- It has a large demand in the current job market
- It is easy to learn and simple to use
- It is open-source and free
- It is secure, fast and powerful
- It has huge community support (tens of millions of developers)
- Java is an object oriented language which gives a clear structure to programs and allows code to be reused, lowering development costs
- As Java is close to C++ and C#, it makes it easy for programmers to switch to Java or vice versa

Java Versions

- https://en.wikipedia.org/wiki/Java_version_history
- <https://www.java.com/releases/>

◆ Java is a platform independent programming language.

Why?

- Java platform independent කියන්නේ platform එකක් මත depend වෙන්නේ නැති (OS එකක් මත බලපාන්නේ නැති) නිසා.



Byte code eka hadana waden thamai java platform independent wenne (.class file eka)

JVM සහ JRE platform-independent ඇ? no

◆ JVM (Java Virtual Machine) – Platform Dependent

JVM එක platform dependent එකක්.

- හැම operating system එකකටම (Windows, Linux, macOS) වෙන වෙනම JVM version එකක් තියෙනවා.
- උදාහරණයක්: Windows වලට .exe JVM එකක්, Linux වලට .bin එකක්.
- ඒක නිසා JVM එක OS එකට අනුව වෙනස් වෙනවා.

→ ඇත්තටම JVM එක OS-specific එකක්.

◆ JRE (Java Runtime Environment) – Platform Dependent

JRE එක තුළට JVM එකත් + libraries + other runtime tools දාගෙන තියෙනවා.

- ඒ වගේම JRE එකත් **platform dependent**.
- JRE එක install කරදීම OS එකට හරිම version එකක් download කරන්න වෙනවා.

→ ඒ නිසා JRE එකත් **platform dependent**.

! නමුත් Java භාෂාව Platform Independent කියන්නේ ඇයි?

එක **Java source code** → **Bytecode** වෙනව නිසා.

- Java compiler එක (`javac`) source code එක compile කරන්නේ **bytecode** විදිහට `.class` file එකක් වෙන්න.
- මේ **bytecode** එක JVM එකක් තියෙන කොහොවන් run කරන්න පූලවන්.
- ඒ නිසා Java කියන භාෂාව **write once, run anywhere (WORA)** කියලා කියනවා.

Component	Platform Independent?	Why?
JVM	✗ No	OS-specific version
JRE	✗ No	Contains platform-specific JVM
Java (bytecode)	✓ Yes	Bytecode can run on any JVM

කෙටියෙන්ම කිවිවේන්:

Java language/platform-independent,
JVM/JRE platform-dependent

● LTS Version

Long-Term Support (LTS) in software refers to a version or release of a software product that is maintained with regular updates, bug fixes, and security patches for an extended period.



(දිගු කාලයක් බය නැතුව use කරන්න පූලුවන් java version වල LTS tag කරල තියෙනව)

● Java SE

Java SE (Java Standard Edition) කියන්නේ Java භාෂාවේ මූලික අනුවාදය. මෙකෙන් තමයි Java වැඩිසටහන් (Programs) සාදන්න සහ ධාවනය කරන්න ඇති මූලික උපාංග සහ මෙවලම් ලැබෙන්නේ (JDK, JVM , JRE ,)

- the standard toolkit for Java programming

Java SE (Standard Edition) is a version of the Java programming language that provides core functionality and libraries for developing desktop, server, and embedded applications.

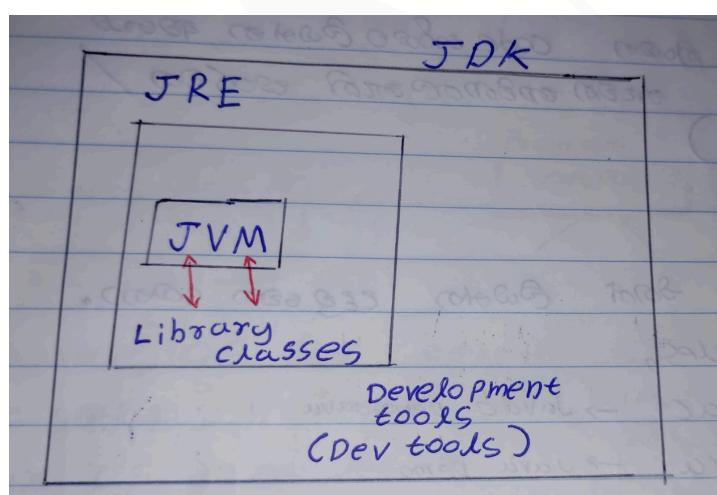
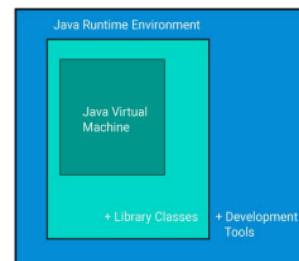


● JDK

It includes everything needed to write, compile, and run java programs.

Java code එකක් ලියන්න සහ compile කරන්න හාවිතා කරන මෙවලම්

The Java Development Kit (JDK) is a software development package used to develop Java applications. It includes everything needed to write, compile, and run Java programs.



● **JRE (Java RunTime Environment)**

- Java Run කරන්න පුළුවන් පරිසරයක් හදනව.

● **JVM**

- **JVM is a virtual machine that enables the execution of Java bytecode**

මෙහේ Java වැඩසටහන ක්‍රියාත්මක කරන "මැෂිනය"

- JVM එක ඇතුළේ තියෙනව interpreter එකක් හෝ JIT compiler එකක්. එයා විසින් Java program එකක් compile කරපු Bytecode එක run time එකෙන් Machine code (binary code) එකක් විදිහට convert කරල Operating System එකට දීල අපට ඕනෑම output එක හදනව.
- Java වලින් හදන program එකක direct output එක JVM තැනුව ගන්න බැව, ape `.class` file eka OS ekata therum ganna amaru wenawa (nathnam bytecode eka run wenne na).

👉 Java program ekak `.java` file → `.class` (bytecode) → **JVM** eken **binary/machine code** → **OS** eka eken **output**.

● **Jit**

The Just-In-Time (JIT) compiler is a component of the runtime environment that improves the performance of Java™ applications by compiling bytecodes to native machine code at run time

● **Library classes**

JDK library classes are pre-built, reusable components that provide ready-to-use functionality for common programming tasks

(Java ඇනුලේ තියෙන code හරියට ලියන්න අවශ්‍ය වන පෙර සාදන ලද නැවත භාවිතා කළ හැකි සංරචක / dependansis)

● Dev Tools

Source code එකක් ලියන්න උදුවූ වෙන කෙනා.
(Java Compiler ඉන්නෙන් මේක ඇනුලේ)

Dev tool examples,

- javac → javac Demo.java (Compile part)
- java → java Demo (Run part)
- jar

● Java compiler

Source code එක Byte code එකක් බවට Convert කරන වැඩි කරනව

● Compiler

Source code එකක් Binary code එකක් බවට Convert කරන වැඩි කරනව.

Byte code සහ Binary code කියන්නේ දෙකක්

- Compiler - translator (english to binary code)

Java compiler saha compiler kiyne 2k java athule wada karanne java compiler (mokada eka platform independent wagma apita byte code ekak haaganna wena nisa), nikaruna compiler use wenne platform dependant C, C++ wage language walata

● Java Interpreter

Byte code එකක් Binary code එකක් බවට Convert කරන වැඩි කරනව.

A program that reads and executes Java Bytecode.

(Interpreter විසින් Byte code එක read කරන්නේ line-by-line)

Let's Install The JAVA addon Into Your Computer.

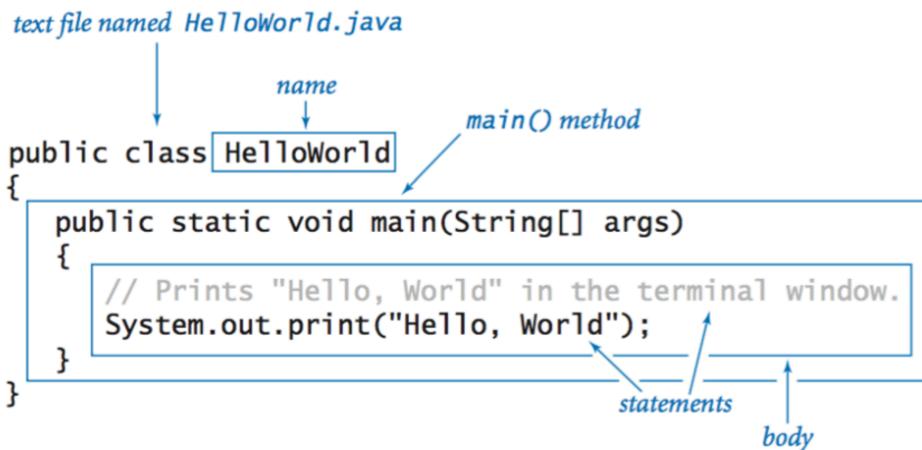
- Verify your **system type** (32bit or 64bit).
- Search **java 8 download** on Google and go to the link.
- Create Oracle account as a student.
- Download the executable file (**.exe**) file.
- Right click on java.exe file and run as administrator. (next > next...)
- Create the System variable.
 - Create **JAVA_HOME**
 - Set the **bin**
- Check java is installed successfully or not in **CMD**
 - **echo %JAVA_HOME%**
 - **java -version**



- **How to create System Variable**
 1. Search bar search “view advanced settings”
 2. Click “Environment Variables”
 3. System variable යටතේ “New” Click kara “JAVA_HOME” variable name එකක්
හදුල value එකට java file එක නියෙන තැන copy කරල paste කරන්න
 4. ඉලගට path එක update කරගන්න ඕනෑම
 5. System variable යටතේ “Path” double click කරන්න
 6. ඉට පස්සේ “New” button එක click කරාම data enter කරන තැනක් එයි, එනනට
“%JAVA_HOME%\bin” දීලෙ ok දෙන්න

- **How to Test JAVA**
 1. Open cmd
 2. Text “**java -version**” enter
 3. Java version ගොඩක් නියෙනව නම් Default select වෙත්ව version එක
සහ ඒ file එක නියෙන තැන “**echo %JAVA_HOME%**” text කරල enter
කරන්න

Hello World Program



String [] args kiyanne variable part ekak, wadiya hithanna epa

Steps for Hello World Program

- Open notepad and write the code(program).
- Save the file as – **HelloWorld.java** .
- Open a command prompt window and go to the directory where you saved the class. Assume it's C:\.
- Type '**javac HelloWorld.java**' and press enter to compile your code.
- Now, type '**java HelloWorld**' to run your program.
- You will be able to see '**Hello World**' printed on the window.

Day - 2

JAVA Mean

OS and Software Developers



C++, kernel is
in C



Objective C,
kernel is in C



C and more..



Windows app
developer



MAC app
developer



Linux app
developer

Programmer.lk

JAVA comes as a Addon Software for all OS`



C++, kernel is
in C



Objective C,
kernel is in C



C and more..

Java Addon
System Softwares



java



java

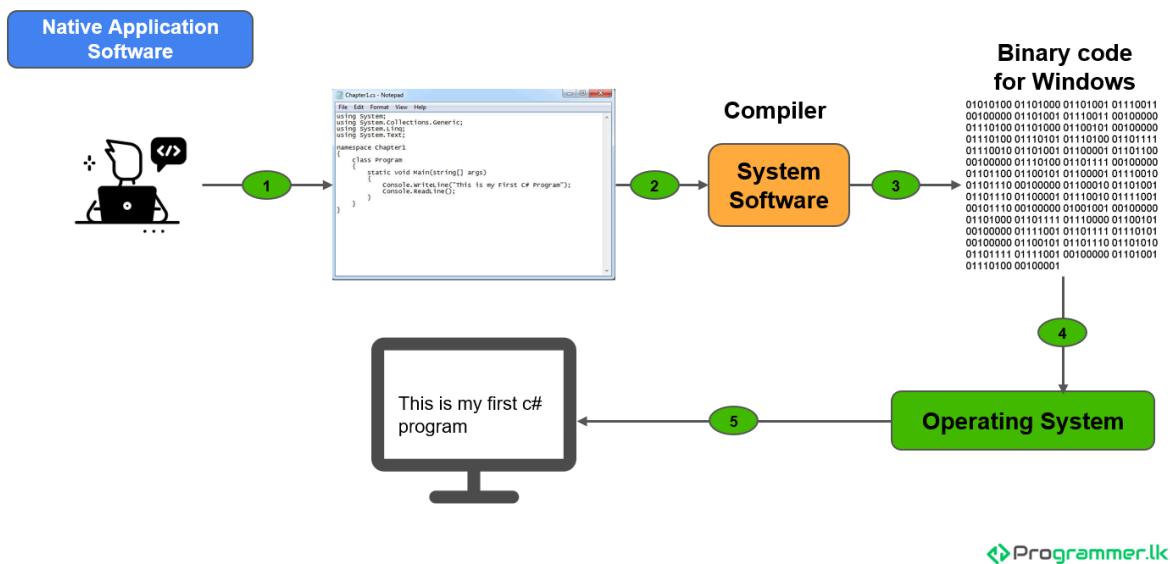


java



Programmer.lk

How a program does work in computer

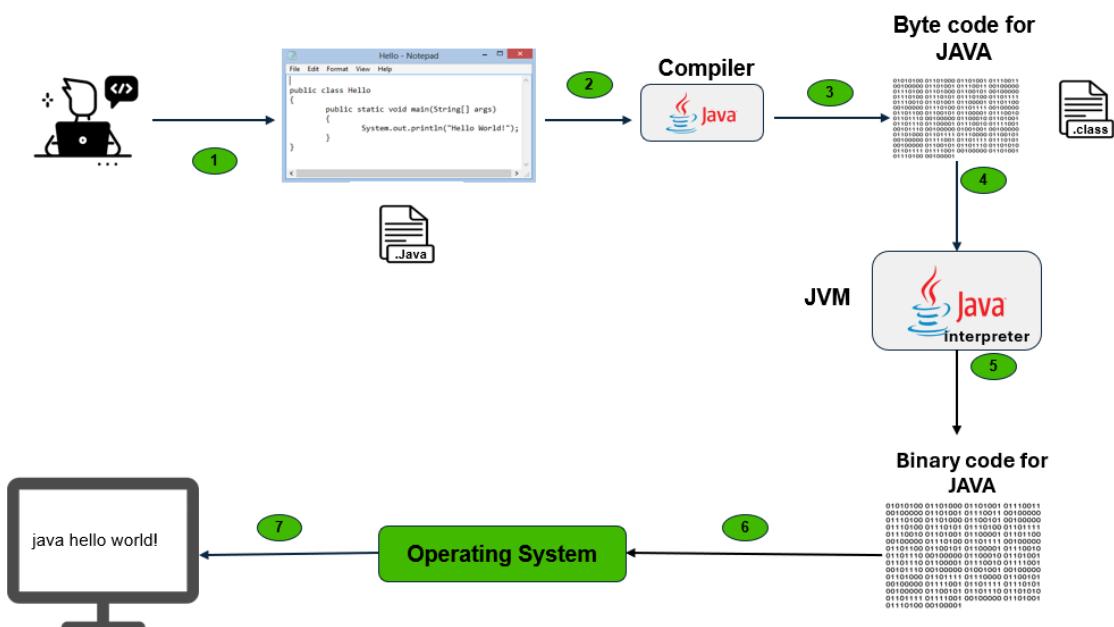


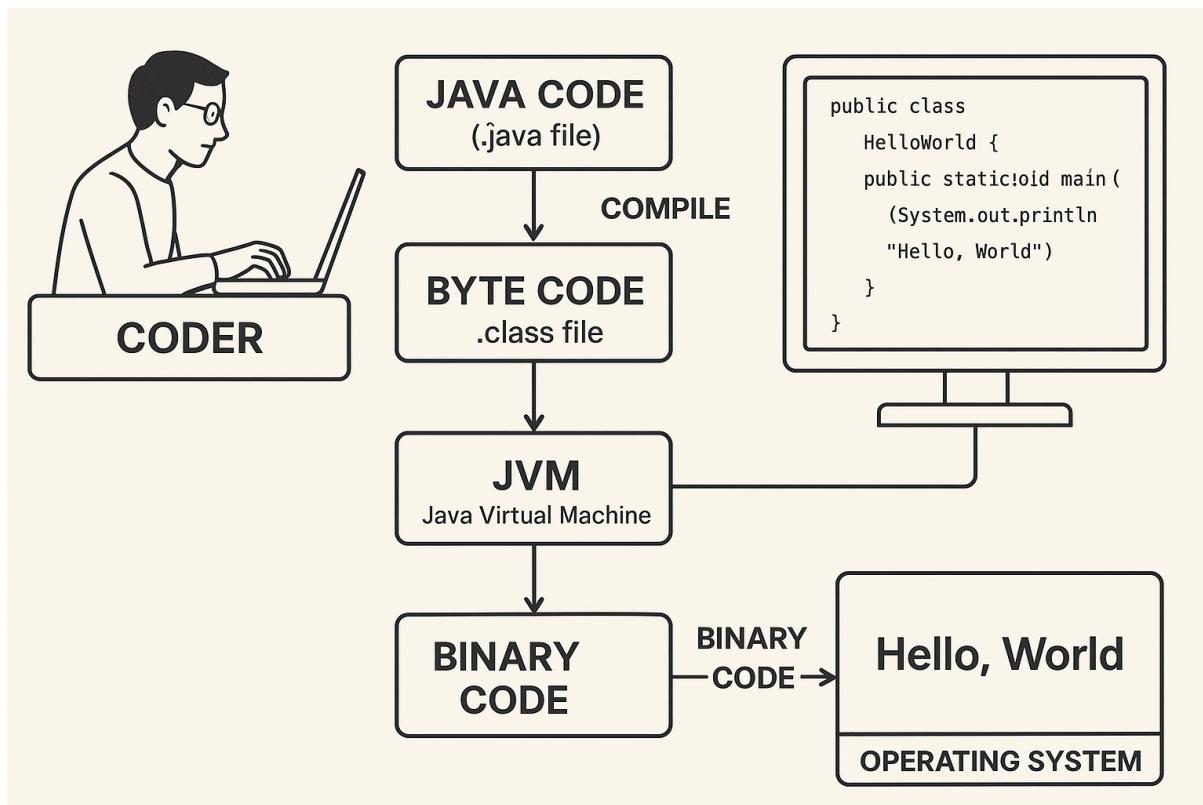
Programmer.lk

- Windows OS එක් ඉදගෙන මේ Program එක හඳුවෙන් මෙයා දුවන්නේ Windows OS එක මත විතරයි.
 - Linux OS එක මත හඳුවෙන් Linux මත විතරයි
 - Mac OS එක මත හඳුවෙන් Mac මත විතරයි
- එම නිසා C, C++ වැනි Languages Platform Dependent Languages වෙනව්

මෙයට විෂයුමක් විදිහට හැම OS එකක් මතම Program එකක් වැඩකරන්න පූලුවන් Platform Independent Language එකක් හඳුව, ඒ Java

How a JAVA program does work in computer





Native application software compiler & java compiler kiyanne dekak , native application software compiler direct human instructions binary code karanawa , namuth java compiler karanne human instruction eka Byte code ekak karana eka, (ema byte code eka binary code ekak karanna jvm thula inna interpreter)

1.Coder (You)

- Writes Java code in a `.java` file.

Example:

```

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

```

2.Java Code (.java file)

- This is the **source code** written in Java.

3,Compile (javac)

- The **Java Compiler (javac)** converts `.java` code into **bytecode**.
- Output: `HelloWorld.class` (bytecode).

4.Byte Code (.class file)

- Platform-independent code that can run on any machine with a JVM.

5.JVM (Java Virtual Machine)

- Loads and interprets the bytecode.
- Converts bytecode into **binary/machine code** at runtime.

6.Binary Code

- Actual **machine-level instructions** understood by the CPU.

7.Operating System (OS)

- Provides the environment for the JVM and manages system resources.

8.Output

- JVM executes the code and interacts with OS to **print Hello, World** on the screen.

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

● Compiler & Interpreter

- Both compilers and interpreters are programs used to translate source code from a high-level programming language into a low-level programming language (like machine code).

Interpreter,

- Interpreter කියන්න් JVM එක ඇතුලේ තියෙන Part එකක්.
- Interpreter විසින් Bytecode එක Read කරන්නේ line by line.
නමුත් Compiler එකටම Read කරල Convert කරනව.

Interpreter විතරක් ඉන්නවනම් ඒක Interpreter Base Language එකක්

- Python

Compiler විතරක් ඉන්නවනම් ඒක Compile Base Language එකක්

- C Language

Interpreter & Compiler දෙකම Base උනු Language එකක් තියෙනව

- Java

Source code -----> Compiler -----> Object code
(High Level Language) (Machine Language / Binary code)

Namuth java compiler gen wenne,
Source code -----> Compiler -----> Object code
(High Level Language) (Byte code)
after

Object code ----->Interpreter -----> Object code
(Byte code) (Machine Language / Binary
code)

Compiler Vs Interpreter

	Compiler	Interpreter
Translation time	Translates all at once	Translates line by line
Speed	Runs faster after compiling	Slower while running
Output	Create a separate file	Does not create a file
Example Languages	C, C++, Java	Python, JavaScript, Java
Error Checking	Shows all errors after compiling	Stops at first error file running

Standard legal code

1

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

◆ Main Method

Legal, Valid main methods

- Source code eka byte code ekakata compile karanawa nam Leagle
Compile error - compile karaddi ena error

- Run karanakota run wenne nathnam invalid
Runtime error - run karaddi ena error

2 (String [] args)

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

3 (String[]args)

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

4 ...

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

```
}
```

5 (String args[]) ,me widihata ... danmoth waradiy

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

6 args wenuwata ona text ekak danna puluwan

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

7 public static maru karala liwwath awulak na

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

```
//Hello Java
```

Legal, Invalid main methods

8 [] missing

```
class Example{
    public static void main(String args){
        System.out.println("Hello Java");
    }
}
```

8 String wenuwata int

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

9 static missing

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

10 public missing

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

11 String[] args missing

```
class Example{  
    public static void main(){  
        System.out.println("Hello Java");  
    }  
}
```

12 Main ----> main

```
class Example{  
    public static void Main(String args[]){  
        System.out.println("Hello Java");  
    }  
}
```

Illegal main Methods

13 string —> String

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

14 void missing

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

15 main missing

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

16 static void maru karala

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

print - thanipeliyata print karanna

println - ilagata print karana eka new line ekaka print karanna

17

```
class Main {
    public static void main(String[] args) {
        System.out.print("Hello, World!");
        System.out.print("Hi, Java");
        System.out.print("Ranil Wikramasinghe");
        System.out.print("Mahindha Rajapaksha");
        System.out.print("Anura Kumara");
        System.out.print("Sajith Premadasa");
    }
}
```

```
//Hello, World!Hi, JavaRanil WikramasingheMahindha RajapakshaAnura KumaraSajith  
Premadasa
```

18

```
class Main {  
    public static void main(String[] args) {  
        System.out.print("ABC");  
        System.out.print("DEF");  
    }  
}  
//ABCDEF
```

19

```
class Main {  
    public static void main(String[] args) {  
        System.out.print("ABC");  
        System.out.println("DEF");  
    }  
}  
//ABCDEF
```

20

```
class Main {  
    public static void main(String[] args) {  
        System.out.println("ABC");  
        System.out.println("DEF");  
        System.out.print("GHI");  
    }  
}  
  
//ABC  
DEF  
GHI
```

21

```
class Main {  
    public static void main(String[] args) {  
        System.out.print("ABC");  
        System.out.println("DEF");  
        System.out.print("GHI");  
    }  
}
```

```
    }  
}  
//ABCDEF  
GHI
```

22

/*
**


```
class Example {  
    public static void main(String[] args) {  
  
        System.out.println(" /\\_/_\\ ");  
        System.out.println(" / o o \\ ");  
        System.out.println(" (   \\" ) ");  
        System.out.println(" \\~(\\" )~\\" );  
    }  
}
```

```
        System.out.println("    \\  /");
        System.out.println("    _|_|_");
    }
}
```

23

```
class Main{
    public static void main(String args[]){
        System.out.println("    *");
        System.out.println("   ***");
        System.out.println("  *****");
        System.out.println("*****");
        System.out.println("  ****");
        System.out.println("   ***");
        System.out.println("    *");
    }
}
```

```
// *
***  
*****  
*****  
****  
***  
*
```

◆ Java Comment

Java comments are notes in the source code that are ignored by the compiler.

- Single Line Comments //comment

```
// print A
```

- Multi Line Comments /* comment */

```
/* System.out.println("A");
System.out.println("B");
System.out.println("C"); */
```

- Doc Comments /** comment */

```
/**  
This is a  
Documentation Comment */
```

Single Line Comments | End of Line Comments

24

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

25

```
class Main{
    public static void main(String args[]){
        //print hello world
        //this is my note
        //those are my notes
        //AMG
        System.out.println("Hello, World!"); //this is my comment
    }
}
```

26

```
class Example{
    public static void main(String args[]) {
        System.out.println("A");
    // System.out.println("B");
        System.out.println("C");
    // System.out.println("D");
        System.out.println("E");
    }
}

//A
C
E
```

27 apata puluhan code ekak hangala thiyanganna run nowena widihata
(comment washayen)

```
class Main{

    public static void main(String args[]) {

        System.out.println("Hello, World!");
        // System.out.println("Hi, Java");

    }
}
```

Multi Line Comments | Block Comments

28

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

        System.out.println("Hello, World!");

        /* this is my book
        this is my cat
        this is my pc
        */

    }
}
```

29

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

```
        System.out.println("A");
/* System.out.println("B");
System.out.println("C");
System.out.println("D"); */
        System.out.println("E");
    }
}

//A
E
```

Document Comment

30

```
class Main{
    /**
     * AMG
     * This is a
     * Documentation Comment
     */

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

//Hello, World!
```

31

```
/*
 * This is a           (mada * auto watenne )
```

```
* Documentation Comment
*/
class Main{
    public static void main(String args[]){
        System.out.println("A");
        System.out.println("B");
        System.out.println("C");
        System.out.println("D");
        System.out.println("E");
    }
}
```

//A

B
C
D
E

```
/***
 * This is a          (mada * auto watenne )
 * Documentation Comment
 */

class Main{
    /**
     * AMG
     * This is a
     * Documentation Comment
     */
}

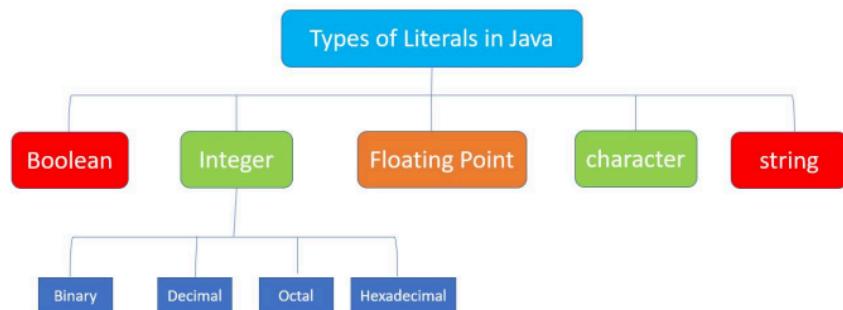
public static void main(String args[]){
    System.out.println("Hello, World!");
}

}
//Hello, World!
```

◆ Java Literals

Literals are data used for representing fixed values.

Literals are fixed values that are directly represented in the source code.



32

```
class Example{
    public static void main(String args[]){
        System.out.println(123); // Integer Literal
        System.out.println(-123); // Integer Literal
        System.out.println(2.45); // Floating point Literal
        System.out.println(-2.45); // Floating point Literal
        System.out.println('A'); // Character Literal
        System.out.println('5'); // Character Literal
        System.out.println('#'); // Character Literal
        System.out.println("ABC"); // String Literal
        System.out.println("123"); // String Literal
        System.out.println(true); // boolean Literal
        System.out.println(false); // boolean Literal
    }
}
```

//123

-123

2.45

-2.45

A

5

#

ABC
123
true
false

● Integer Literals

Integer literals in Java are whole number values written directly in the code, without any decimal points.

These literals represent constant integer values and can be written in different number systems (bases).

- **Binary (0B or ob)** 2 paada (0,1)
- **Octal (0)** 8 paada (0,1,2,3,4,5,6,7)
- **Decimal (No pref)** 10 paada (0,1,2,3,4,5,6,7,8,9) {ALL}
- **HexaDesimal (0x)** 16 paada (0,1,2,3,4,5,6,7,8,9, A, B, C, D, E, F)
(0,1,2,3,4,5,6,7,8,9, a, b, c, d, e, f)

10 11 12 13 14 15

Api me mona wargaya print karath apata answer eka enne Decimal value eka widihata

Binary

33

```
class Example{
    public static void main(String args[]){
        System.out.println(1010); // (Decimal) prints 1010
        System.out.println(0b1010); // (Binary) prints 10
        System.out.println(0B1010); // (Binary) prints 10
    }
}

//1010
10      1x2^3+ 0x2^2+1x2^1+ 0x2^0=+0+2+0
10
```

Octal

34

```
class Example{  
    public static void main(String args[]){  
        System.out.println(144); // (Decimal) prints 144  
        System.out.println(0144); // (Octal) prints 100  
        // System.out.println(0148); // Illegal (Octal literal ends at 8, 9  
valid na)  
    }  
}  
//144  
100 1x8^2 + 4x8^1 + 4x8^0=64+32+4 =100
```

Decimal

35

```
class Main {  
    public static void main(String[] args) {  
        System.out.println(2); // prints 2  
        System.out.println(3); // prints 3  
        System.out.println(4); // prints 4  
        System.out.println(10); // prints 10  
        System.out.println(20); // prints 20  
        System.out.println(200); // prints 200  
        System.out.println(132412312); // prints 132412312  
        // all decimal  
    }  
}  
//2  
3  
4  
10  
20  
200  
132412312
```

HexaDecimal

36

```
class Example{
    public static void main(String args[]){
        System.out.println(64); // (Decimal) prints 64
        System.out.println(0x64); // (Hexa-Decimal) Prints 100
        System.out.println(0x1E); // (Hexa-Decimal) prints 30
    }
}

//64
100
30
```

37 HexaDecimal agayan wana A-F sadaha a-f simple agayan uwada baawithaa karanna puluhan, x wenuwata X da baawithaa karanna puluhan

```
class Example{
    public static void main(String args[]){
        System.out.println(0x1E); // prints 30
        System.out.println(0x1e); // prints 30
        System.out.println(0X1E); // prints 30
    }
}

//30
30
30
```

38

```
class Example{
    public static void main(String args[]){
        System.out.println(100); //decimal
        System.out.println(10); //decimal
        System.out.println(111111); //decimal
        System.out.println(1010); //decimal
        System.out.println(0b1010); //binary
        System.out.println(0B1010); //binary
        //System.out.println(0B1012); //illegal (binary 0, 1 pamanai 2 not valid)
        System.out.println(01010); //octal
        System.out.println(0x1010); //HexaDesimal

    }
}

//100
10
111111
1010
10
10
520
```

4112

39

```
class Example{
    public static void main(String args[]){
        System.out.println(144); //Decimal-> 144
        System.out.println(0144); //octal -> 100
        // System.out.println(0148); //illegal 8 octal walata yedenne nathi
nisa
    }
}

//144
100
```

40

```
class Main {
    public static void main(String[] args) {
        System.out.println(2); // prints 2
        System.out.println(3); // prints 3
        System.out.println(4); // prints 4
        System.out.println(10); // prints 10
        System.out.println(20); // prints 20
        System.out.println(200); // prints 200
        System.out.println(132412312); // prints 132412312
        //all decimal
    }
}

//2
3
4
10
20
200
132412312
```

41

```
class Main {  
    public static void main(String[] args) {  
        System.out.println(64); // Decimal  
        System.out.println(0x64); // HexaDecimal  
    }  
}  
  
//64  
100
```

42

```
class Example{  
    public static void main(String args[]){  
        System.out.println(10_000_000); //prints 10000000  
    }  
}  
  
//10000000
```

Decimal to Binary convert

```
import java.util.Scanner;  
class Main {  
    public static void main(String[] args) {  
        Scanner input = new Scanner(System.in);  
  
        System.out.print("Enter a number: ");  
        int num = input.nextInt();  
  
        String num1 ="";  
  
        while(num > 0){  
            int rem = num % 2;  
            num = num / 2;  
  
            num1 = rem + num1;  
        }  
        System.out.print(num1);  
    }  
}
```

Day - 3

• Floating-Point Literals

Floating-point literals are numeric values with a decimal point or written in exponential (scientific) notation.

They are used to represent non-integer (fractional) numbers.

Examples:

3.14 - a double literal

2.0 - another double literal

0.5f - a float literal (note the 'f' suffix)

1.23e2 - exponential notation (equals 123.0)

6.02E23d - exponential notation (note the 'd' suffix)

$$1.23\text{e}2 = 1.23 * 10^2$$

Floating Point Literals

Float

(Single Precision)

32 bits

Accuracy- Low

Memory storage - Low

E.g- 1.52f

6.325f

97.53f

Double

(Double Precision)

64 bits

Accuracy- High

Memory Storage- High

E.g- 1.532

2.433e2

1.75e2d

- 1.532f වැනි අවසානයට f වැටෙන දූෂම සංඛ්‍යා ඇරෝන්න අනෙකුත් සියලුම සංඛ්‍යා Double ගනයට ඇයන් වේ.
- ape pariganaka wala dashama sankya default enne Double sankyaa widihata.
- Memory eka gana worry wenne nathnam Double thoraganna.
- Api memory eka gana worry wenawa nam 1.532 wage sankyaawak awoth danna hodama thana float.

43

```
class Example{
    public static void main(String args[]){
        System.out.println(11.223);
        System.out.println(12.0000);
        System.out.println(1.2E4); // 1.2 x 10^4 = 1.2 x 10000 = 12000
        System.out.println(1.2e-3); // 1.2 x 10^-3 = 1.2/ 1000 = 0.0012
        System.out.println(1.23f); // (float)
    }
}
```

11.223

12.0

12000.0

0.0012

1.23

•Character Literals

Character Literals

Unicode is an international character encoding standard that provides a unique number for every character across languages and scripts, making almost all characters accessible across platforms, programs, and devices.

A character literal in Java is a **single Unicode character written inside single quotes (' ')**.

'A'

- Unicodes- Characters computer eka haduna ganna code
- 😊emoji ekak unath character ekak

44

```
class Example{
    public static void main(String args[]){
        System.out.println('A'); // (Character)
        System.out.println("A"); //this not character (String)
        // System.out.println('Hello'); // illegal
    }
}
```

A
A

•String Literals

A String literal in Java is a **fixed text value written directly in double quotes (" ")**.

“Hello World!”

(Text liyanna use karanawa string)

45

```
class Example{
    public static void main(String args[]){
        // System.out.println('Hello'); // illegal
        System.out.println("Hello"); //String
        System.out.println("123"); //String
    }
}
```

Hello

123

•Boolean Literals

Boolean literals **represent only two values true or false.**

true false

Boolean kiyala aduraganna nam **true & false** simple wiya yuthui

- **True** ✗ -----> true ✓
- **False** ✗ -----> false ✓

50

```
class Example{
    public static void main(String args[]){
        System.out.println(true); // true
        System.out.println(false); // false
        //System.out.println(True); // illegal T->t
        //System.out.println(False); // illegal F->f
    }
}
//true
false
```

◆ Escape Characters

Escape characters in Java are special symbols starting with \ used inside strings to represent things like:

- \n → new line
- \t → tab
- \" → double quote
- \\ → backslash

Escape characters in Java are special sequences of characters used to represent characters that are hard to type directly or have a special meaning, like new lines, tabs, quotes, etc.

They always start with a backslash (\).

- \" = " → " \" hello \" " -> "hello"
- \\ = \
- \f new java version walata wada na
‐ back slash
‐ forward slash

\t	Inserts a tab
\b	Inserts a backspace
\n	Inserts a newline
\r	carriage return. ()
\f	form feed
'	Inserts a single quote
"	Inserts a double quote
\	Inserts a backslash

List of Escape Sequences in Java

(\f - pages Break)

46

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

        //System.out.println("hello"); // (error) i want print "hello"
        System.out.println("\\"hello\\\""); // print "hello"
        System.out.println("jav\ba"); //print jaa
        System.out.println("ab\tcd"); //print ab      cd
        System.out.println("3\5"); // print 3 (i want print 3\5)
        System.out.println("3\\5"); //print 3\5
```

```
    }
}

"hello"
jaa
ab      cd
3
3\5
```

47

```
class Example{
    public static void main(String args[]){
        System.out.println("\n\\\\\\\\\\\\\\\\");
    }
}
"\\\\"
```

48

```
class Example{
    public static void main(String args[]){
        System.out.println("C:\\\\\\user\\\\home\\\\java\\\\bin");
    }
}
C:\\user\\home\\java\\bin
```

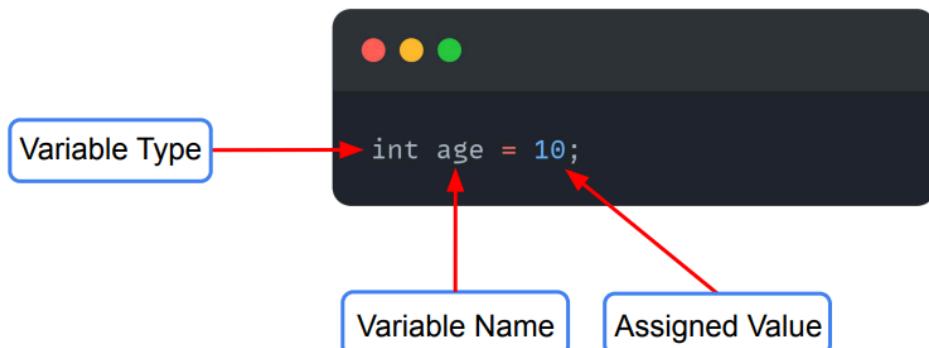
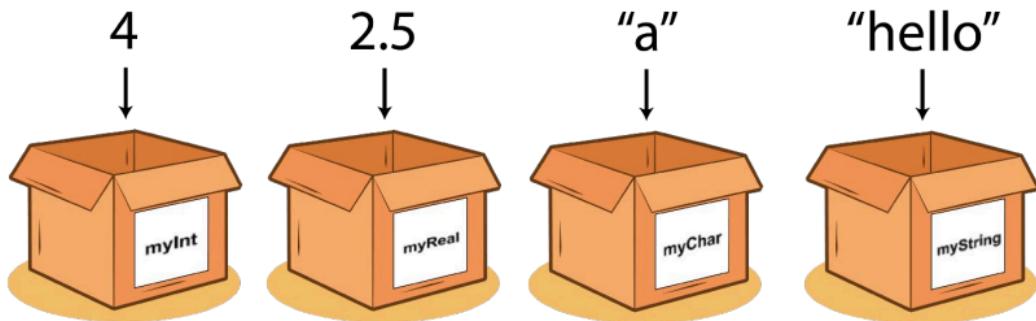
49

```
class Example{
    public static void main(String args[]){
        System.out.println("abc\\ndef"); // \n = new line
    }
}
abc
Def
```

◆ Variable

Nam karapu storage location ekaka thiyyena value or data ekak

Variable in Programming is a named storage location that holds a value or data.



Variable Type

|

Int age = 20; ←Assign Value(Data type)

|

Variable Name

VARIABLE TYPE

- Variable එක හාවිනා වන්නේ කොහොද, කොහොමද කියන එක
- A **variable type** refers to **how** and **where** a variable is used in a program — its scope, lifetime, and visibility
- Variable type එක කියන්නේ variable එකක් memory එකේ representation එක, scope එක, lifetime එක වගේ technical ලක්ෂණ පදනම් කරගෙන panthi කරණය කරන ක්‍රමයකි
- How/where the variable is used in the program

Rules of Variable Name

1. A variable name can consist of Capital letters A-Z, lowercase letters a-z digits 0-9, and two special characters such as _ underscore and \$ dollar sign.
2. The first character must not be a digit.
3. Blank spaces cannot be used in variable names.
4. Variable names are case-sensitive.
5. Java keywords cannot be used as variable names.



4. Case - sensitive?

```
class Main {  
    public static void main(String[] args) {  
  
        int hi = 53;  
        int Hi = 76;  
  
        System.out.println(hi);  
  
    }  
}  
//53
```

Variable Steps

1. Define a Variable named Z with the Variable type int.
2. Assign Value 800 to that Variable.
3. Print the Variable Z.

```
class Example{  
    public static void main(String args[]) {  
  
        int z = 800;  
    }  
}
```

```
        System.out.println(z); // 800
    }
}
//800
```

51

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

        int age = 20;

        System.out.println(age);
    }
}
```

20

52

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

        int age;
        age = 100;

        System.out.println(age);
    }
}
```

100

53

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

        int x;
        x = 100;

        System.out.println(x); // 100
    }
}
```

```
        System.out.println('x') ; // x (character)
        System.out.println("x") ; // x (String)
    }
}
100
x
x
```

54

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

        int x;
        x = 100;

        System.out.println(x) ; //100

        x = 200;

        System.out.println(x) ; //200
    }
}
```

100

200

55

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

        int x;
        x = 100;
        x = 200;

        System.out.println(x);
    }
}
```

200

56

```
class Example{
    public static void main(String args[]){
        int x;
        x = 200;
        System.out.println(x); //200

        int y;
        y = 500;
        System.out.println(y); //500

        int z;
        z = 800;
        System.out.println(z); //800
    }
}
200
500
800
```

57

```
class Example{
    public static void main(String args[]){
        int x,y,z;
        x = 200;
        y = 500;
        z = 800;
        System.out.println(x); //200
        System.out.println(y); //500
        System.out.println(z); //800
    }
}
200
500
800
```

Method mean -> main method part eka

58

Eka method ekak athule Hadapu variable ekak nawatha hadanna ba

```
class Example{
```

```
public static void main(String args[]){  
  
    int x,y,z;  
    x = 200;  
    y = 500;  
    z = 800;  
    System.out.println(x);  
    System.out.println(y);  
    System.out.println(z);  
  
    int a;  
    a = 100;  
  
    // int x; // (illegal) hadapu variable ekak aye hadanna ba  
    // variable y is already defined  
}  
}  
200  
500  
800
```

59

```
class Example{  
    public static void main(String args[]){  
  
        int x=1000,y=2000,z=3000;  
  
        System.out.println(x); //1000  
        System.out.println(y); //2000  
        System.out.println(z); //3000  
  
    }  
}
```

1000
2000
3000

60

```
class Example{  
    public static void main(String args[]){  
  
        int x=1000,y,z=3000;  
  
        System.out.println(x); //1000
```

```
//System.out.println(y); //error y ta value ekak assign karala nathi  
nisa  
    System.out.println(z); //3000  
}  
}  
1000  
3000
```

61

```
class Example{  
    public static void main(String args[]){  
  
        int x=1000,y,z=3000;  
  
    }  
}
```

No problem, e kiyanne y assing karanne nathuwa print karanna giyoth thamai awula wenne

62

```
class Example{  
    public static void main(String args[]){  
  
        int x=1000,y,z=3000;  
  
        y = 5000;  
  
        System.out.println(x); //1000  
        System.out.println(y); //5000  
        System.out.println(z); //3000  
  
    }  
}
```

1000
5000
3000

63

```
class Example{  
    public static void main(String args[]){  
  
        int x,y,z;  
        /*x = 1000;  
        y = 1000;  
        z = 1000; */  
    }  
}
```

```

x = y = z = 1000;

System.out.println(x);
System.out.println(y);
System.out.println(z);

}

}

```

1000
1000
1000

64

x,y,z ta wena wenama variable widihata 3000 type wenna ona nam

```

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

        //int a = 100;

        //int x,y,z = 3000;    //(wrong) methanadi wenne z wala agaya
witharak gannawa anith dekata assign karala na wage penawa
        //int x=y=z=3000;      //(wrong) methanadi wenne x wala agaya
witharak gannawa anith dekata assign karala na wage penawa


        int x,y,z;
/*x = 3000;
y = 3000;
z = 3000; */

        x = y = z = 3000;

        System.out.println(x);
        System.out.println(y);
        System.out.println(z);

    }
}

```

3000
3000
3000

◆ String concatenation & arithmetic addition

(string ekathu karanawa) & (gaanak ekathu karanawa)

● String concatenation

Join two or more String together

Use +

65

```
class Example{
    public static void main(String args[]){
        System.out.println("Hello" + "India"); // i want bind "Hello" to
        "India"
        System.out.println("Hello" + "India" + "Singapore" + "Japan" );
        System.out.println("Hello " + "India " + "Singapore " + "Japan" );
        //i want Space
    }
}
```

HelloIndia

HelloIndiaSingaporeJapan

Hello India Singapore Japan

● arithmetic addition

Combining two or more numbers to get their total and sum

66

```
class Example{
    public static void main(String args[]){
        System.out.println(20 + 20); // 40
        System.out.println(12 + 120); // 132
        System.out.println(10 + 20 + 30); // 60
    }
}
```

```
    }
}
```

67

String + Number = String

```
class Example{
    public static void main(String args[]){
        System.out.println(20 + 20); // 40
        System.out.println("10" + "20"); // 1020 (String)
        System.out.println("10" + 20); // 1020 (String)
    }
}
```

40

1020

1020

1020

68

Numbers kiyawanne —> paththata ekai mehema wenne

```
class Example{
    public static void main(String args[]){
        System.out.println(20 + 30 + "10"); // 5010
        System.out.println("20" + 30 + 10); // 203010 (String)
    }
}
```

5010

203010

69

```
class Example{
    public static void main(String args[]){
        int x = 10;
        int y = 20;
        int z = 30;

        System.out.println(x + y); // 30
    }
}
```

```
        System.out.println(x + y + z); // 60
        System.out.println("x + y + z"); // x + y + z
        System.out.println("x + y" + z); // x + y30
        System.out.println("x" + y + z); // x2030
        System.out.println(x + y + "z"); // 30z
        System.out.println(x + "y" + z); // 10y30

    }
}

30
60
x + y + z
x + y30
x2030
30z
10y30
```

70

```
class Example{
    public static void main(String args[]){
        int age = 20;

        System.out.println("Your age: "+ age); // Your age: 20
    }
}
```

Your age: 20

71

```
class Example{
    public static void main(String args[]){
        int year = 2025;
        int month = 5;
        int day = 14;

        System.out.println(year + "/" + month + "/" + day); // 2025/5/14
    }
}
```

2025/5/14

◆Java Keyboard Inputs

This is the process of reading user input from the keyboard during program execution.

It is commonly handled using the **Scanner** class from the `java.util` package.



Use **Scanner** for Keyboard Inputs

```
public static void main(String[] args) {  
    Scanner input = new Scanner(System.in); // Create a Scanner object  
  
    System.out.println("Enter your age: ");  
    int age = input.nextInt(); // Read user input  
  
    System.out.println("Your age is: " + age); // Output user input  
}
```

Input Types

<code>nextInt()</code>	Reads a int value from the user
<code>nextByte()</code>	Reads a byte value from the user
<code>nextShort()</code>	Reads a short value from the user
<code>nextLong()</code>	Reads a long value from the user
<code>nextDouble()</code>	Reads a double value from the user
<code>nextFloat()</code>	Reads a float value from the user
<code>nextLine()</code>	Reads a String value from the user
<code>nextBoolean()</code>	Reads a boolean value from the user



String-> `next()`

Char -> `next().charAt(0)`

72 (input name eka variable name ekak eka wenas karanna puluwan)

```
import java.util.Scanner; // java wala utile folder eke util  
file eka thule thiyyena scanner methanata import karanawa  
kiyana adahasa  
  
class Main{  
    public static void main(String args[]){  
        Scanner input = new Scanner(System.in); //create a scanner  
object  
        int x = input.nextInt(); //read user input enter karanna  
opportunity eka hadnawa xta e value eka dala eka print  
karanawa  
        System.out.println(x);  
    }  
}
```

2000

2000

Scanne wenuwata * danmama Scannerwitharak nemei ethakota all import
karagannawa,arrays unath gahanna puluwan nawatha import nokara

```
import java.util.*; /*all eka gannawa , kalin kale Scanner witharak  
gaththa eka  
  
class Main{  
    public static void main(String args[]){  
        Scanner input = new Scanner(System.in);  
        int x = input.nextInt(); //enter karanna opportunity eka hadnawa xta e  
value eka dala eka print karanawa  
        System.out.println(x);  
    }  
}
```

73

```
import java.util.Scanner;

class Main{
    public static void main(String args[]){
        Scanner input = new Scanner(System.in);
        int x = input.nextInt(); //enter karanna opportunity eka hadnawa xta e
value eka dala eka print karanawa
        System.out.println("Value: " + x); //Value: 5000
    }
}
```

5000

Value: 5000

74

```
import java.util.Scanner;

class Main{
    public static void main(String args[]){
        Scanner input = new Scanner(System.in);
        System.out.println("Enter number: ");
        int x = input.nextInt();
        System.out.println("Value: " + x); //Value: 5000
    }
}
```

75

```
import java.util.Scanner;

class Main{
    public static void main(String args[]){
        Scanner input = new Scanner(System.in);

        System.out.println("Enter number 1: ");
        int x = input.nextInt();

        System.out.println("Enter number 2: ");
        int y = input.nextInt();

        System.out.println("Num 1: " + x);
        System.out.println("Num 2: " + y);
    }
}
```

```
}
```

Enter number 1:

20

Enter number 2:

70

Num 1: 20

Num 2: 70

76

```
import java.util.Scanner;

class Main{
    public static void main(String args[]){
        Scanner input = new Scanner(System.in);

        System.out.println("Enter number 1: ");
        int x = input.nextInt();

        System.out.println("Enter number 2: ");
        int y = input.nextInt();

        System.out.println("Sum: " + x + y);
    }
}
```

Enter number 1:

67

Enter number 1:

56

Sum: 6756 (It's not real answer)

77

```
import java.util.Scanner;

class Main{
    public static void main(String args[]){
        Scanner input = new Scanner(System.in);

        System.out.println("Enter number 1: ");
        int x = input.nextInt();

        System.out.println("Enter number 1: ");
        int y = input.nextInt();

        System.out.println("Sum: " +( x + y));
    }
}
```

```
}
```

Enter number 1:

76

Enter number 1:

85

Sum: 161

78

```
import java.util.Scanner;

class Main{
    public static void main(String args[]){
        Scanner input = new Scanner(System.in);

        System.out.println("Enter number 1: ");
        int x = input.nextInt();

        System.out.println("Enter number 2: ");
        int y = input.nextInt(); //input

        int z = x + y ;

        System.out.println("Sum: " + z);
    }
}
```

Enter number 1:

5

Enter number 2:

6

Sum: 11

79

```
import java.util.Scanner;

class Main{
    public static void main(String args[]){
        Scanner input = new Scanner(System.in);

        System.out.println("Enter number 1: ");
        int x = input.nextInt();

        System.out.println("Enter number 1: ");
        int y = input.nextInt();

        x = x + y;

        System.out.println("Sum: " + x);
    }
}
```

```
    }  
}
```

Enter number 1:

5

Enter number 1:

7

Sum: 12

♣Agayan maru karana widiha (Swap)

swap=numbers maru karanawa

80

Num1 , num 2 input maru wela answer eka widihata code eka gahanna

```
import java.util.Scanner;  
  
class Main {  
    public static void main(String[] args) {  
        Scanner input = new Scanner(System.in);  
  
        System.out.println("Enter number 1: ");  
        int num1 = input.nextInt();  
  
        System.out.println("Enter number 2: ");  
        int num2 = input.nextInt();  
  
        num1 = num1 + num2;  
        num2 = num1 - num2;  
        num1 = num1 - num2;  
  
        System.out.println("Num1 = " + num1);  
        System.out.println("Num2 = " + num2);  
    }  
}
```

Enter number 1:

4

Enter number 2:

6

Num1 = 6

Num2 = 4

Using temp

```
import java.util.Scanner;
```

```
class Main {  
    public static void main(String[] args) {  
        Scanner input = new Scanner(System.in);  
        System.out.println("Enter number 1: ");  
        int num1 = input.nextInt();  
        System.out.println("Enter number 2: ");  
        int num2 = input.nextInt();  
        /////////////////  
        int temp = num1;  
        num1 = num2;  
        num2 = temp;  
        /////////////////  
        System.out.println("Num1 = " + num1);  
        System.out.println("Num2 = " + num2);  
    }  
}
```

Enter number 1:

600

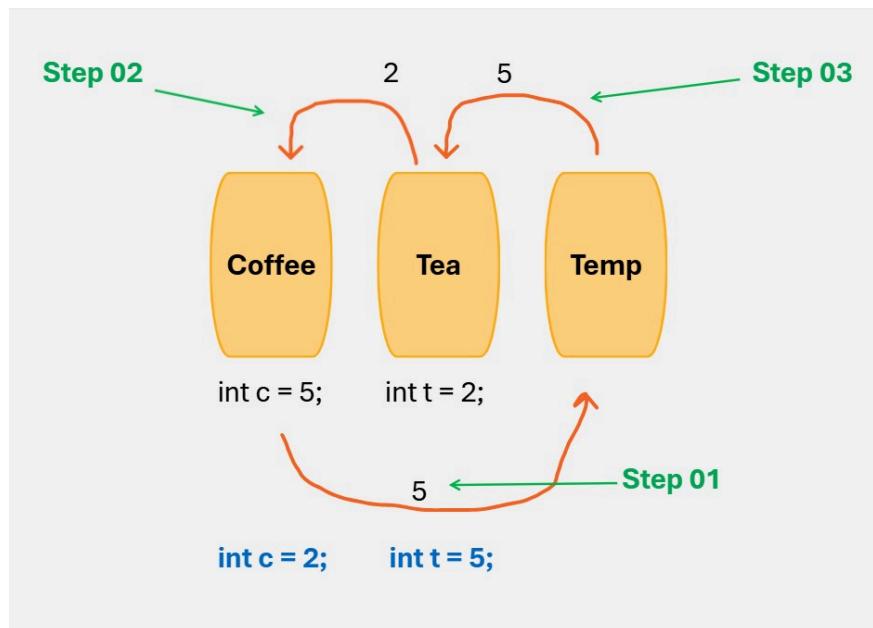
Enter number 2:

58

Num1 = 58

Num2 = 600

Coffee cup ekak saha tea ekaka thiyena ewa thiyena cup deka maaru karaganna
ona, api ekata thawa tempury cup ekak use karanawa



```

import java.util.Scanner;

class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.println("Enter number 1: ");
        int num1 = input.nextInt();
        System.out.println("Enter number 2: ");
        int num2 = input.nextInt();
        /////////////////
        num1=num1+num2-(num2=num1);

        ///////////////
        System.out.println("Num1 = " + num1);
        System.out.println("Num2 = " + num2);
    }
}
  
```

$$10+20-(y=10)$$

$$30-10=20$$

Day - 3

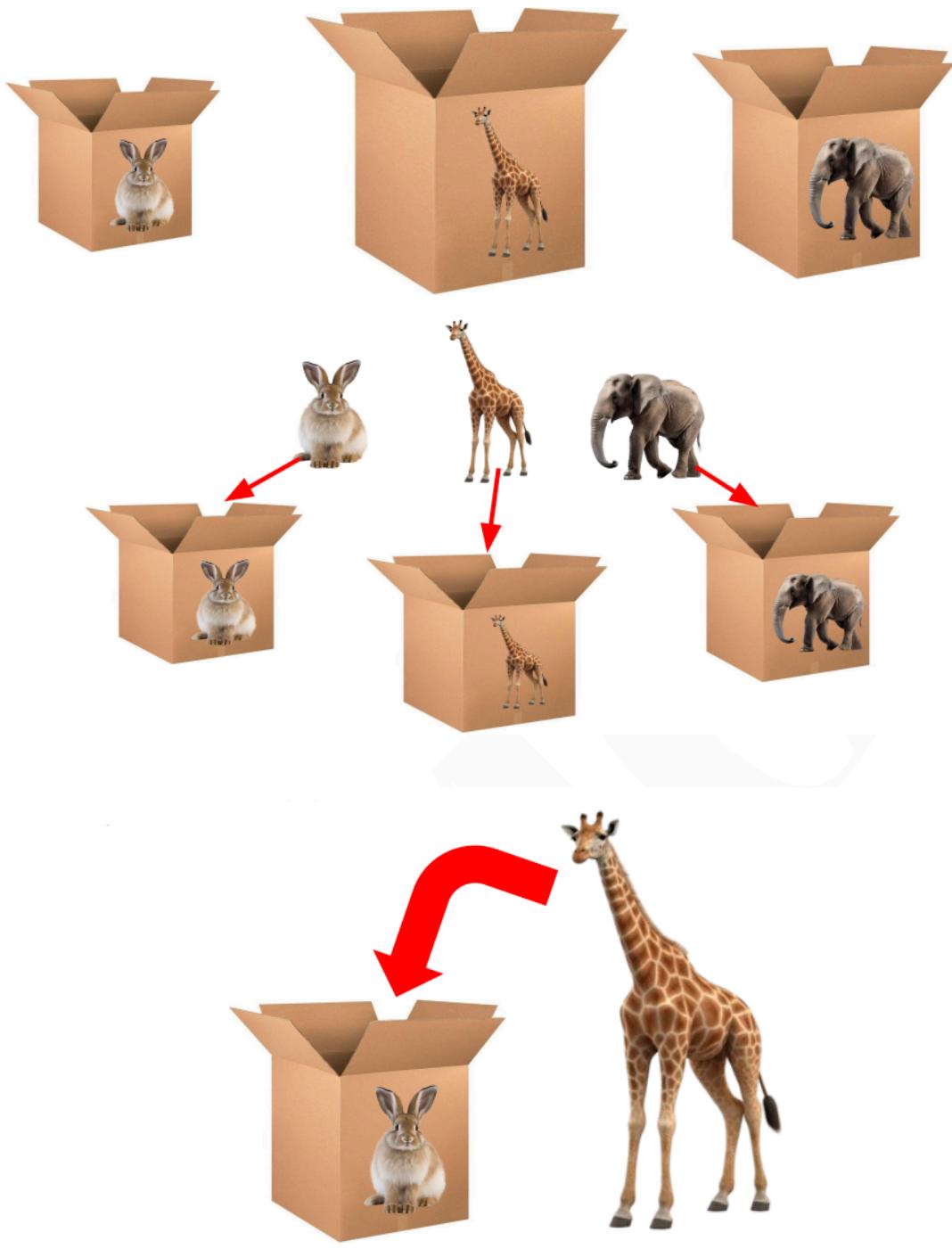
◆ DATA TYPES

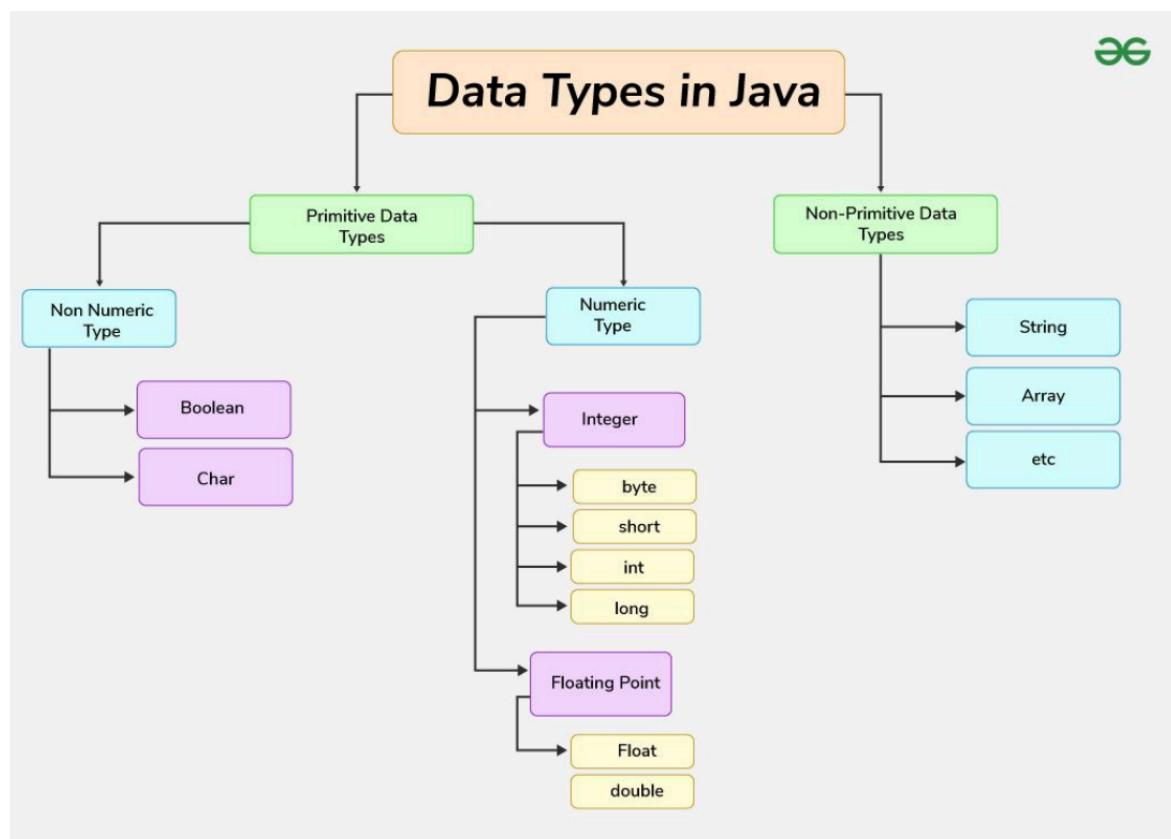
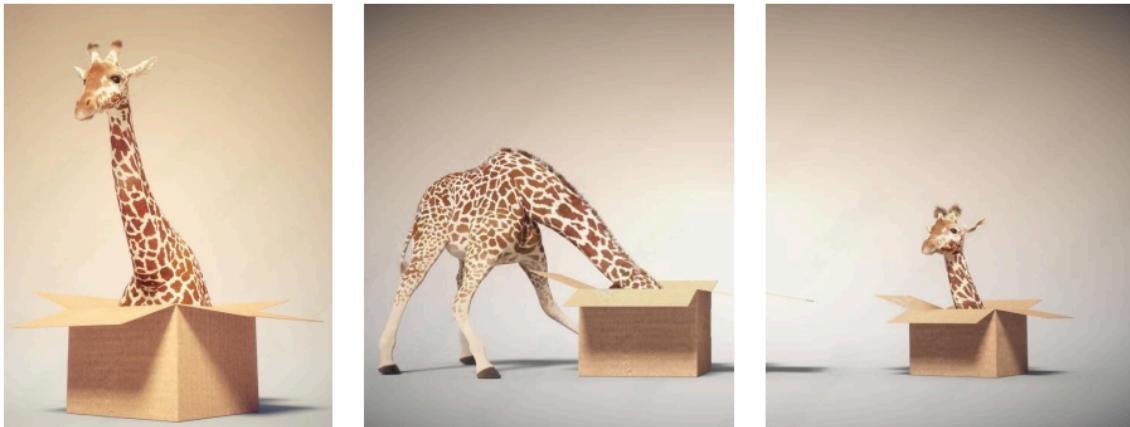
- Variable එකට assign කරන දත්තවල වර්ගය
- A **data type** defines the **type of data** a variable can hold
- Data type එක කියන්නේ variable එකක තබාගන්න දත්ත වර්ගය (type of data) දක්වන එකක්.
-  Data type එකෙන් කියන්නේ "මොන වගේ දත්තයක්ද මේක?" කියන එකයි
- What type of data the variable holds

Data types define the type of data that a variable can store.



Defines the scope and behavior of the variable.





● Primitive Data Types

Simple definition for the primitive data types,

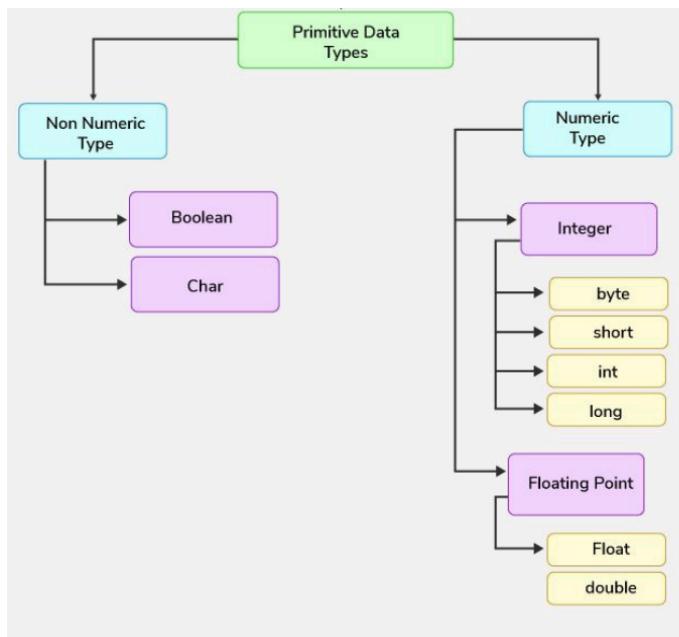
Primitive Data Types

Basic types that hold one simple value, like a number, letter, or true/false.



- Simple single value data type
- Primitive data types are the fundamental, built-in data types that a language uses to represent simple values like numbers, characters, and boolean values.
- Primitive data types are the basic building blocks of data in a program.

```
class Main {  
    public static void main(String[] args) {  
        int x = 10; // int  
        System.out.println(x);  
        // x = 1.5; // is double type,error:incompatible type  
        // (illegal)  
    }  
}
```



Integer

Byte

```

class Main {

    public static void main(String[] args) {

        byte b = 5;

        b = 100;

        b = 127;

        //b = 128; // -128 to 127

        System.out.print(b);

    }

}
  
```

127

```

class Main {
    public static void main(String[] args) {
        byte b1 = 5;
        b1 = 100;
        b1 = 127; // Max Value
        b1 = -128; // Min Value
        //b1 = 128; // illegal (because the range is -128 to 127)
        System.out.print(b1);
    }
}
  
```

```
}
```

-128

```
class Main {
    public static void main(String[] args) {
        byte b = 5;
        b = 100;
        b = 127; // Max Value
        //b = 128; // error:incompatible type ( {Min}-128 to {Max}127 )
        b = -1;
        b = 125;
        b = -128; // Min Value
        // b = -180; // error:incompatible type ( -128 to 127 )
        // b = 150; // error:incompatible type ( -128 to 127 )
        System.out.print(b);
    }
}
```

Short

```
class Main {
    public static void main(String[] args) {
        short s1 = 5;
        s1 = 100;
        s1 = 127;
        s1 = 128;
        s1 = 32767; // Max value
        // s1 = 32768; // illegal - :incompatible types
        s1 = -32768; // Min value
        // s1 = -32769; // illegal - :incompatible types
        System.out.println(s1);
    }
}
```

-32768

int

```
class Main {  
    public static void main(String[] args) {  
  
        int i = 10;  
        System.out.println(i);  
  
        System.out.println(Integer.MAX_VALUE); // 2147483647  
        System.out.println(Integer.MIN_VALUE); // -2147483648  
  
    }  
}
```

10
2147483647
-2147483648

```
class Main {  
    public static void main(String[] args) {  
        int i = 10;  
        i = 200;  
        System.out.println(i);  
  
        i = 2147483647;  
        // i = 2147483648; // error: integer number too large  
  
        System.out.println(Integer.MAX_VALUE); // 2147483647  
        System.out.println(Integer.MIN_VALUE); // -2147483648  
  
        System.out.println(i);  
    }  
}
```

200
2147483647
-2147483648
2147483647

Long

```
class Main {
```

```
public static void main(String[] args) {  
    // byte, short  
    long l = 10;  
    System.out.println(l);  
  
    System.out.println(Long.MAX_VALUE); // 9223372036854775807  
    System.out.println(Long.MIN_VALUE); // -9223372036854775808  
}  
}  
10  
9223372036854775807  
-9223372036854775808
```

♣Max & Min

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

```

        // byte, short
System.out.println(Byte.MAX_VALUE); // 127
System.out.println(Byte.MIN_VALUE); // -128

System.out.println(Short.MAX_VALUE); // 32767
System.out.println(Short.MIN_VALUE); // -32768
}

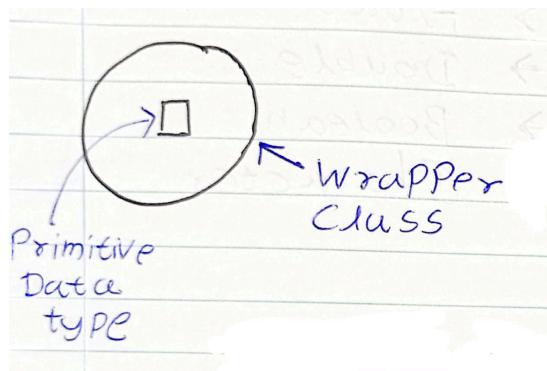
}

```

127
-128
32767
-32768

(Short.MAX_VALUE) (Short.MAX_VALUE)
 (Byte.MAX_VALUE) (Byte.MAX_VALUE)
 |
 Wrapper Class

Wrappe = ඔනල



- අපි class එකෙන් තමයි max value එක ඉල්ලන්නේ. එක class එකට නියෙන හැකියාවක් (අපට direct හිහිල්ල primitive data type එකෙන් මේක ඉල්ලන්න බැවු)
- Wrapper class වලින් primitive data type වල උපරිමය සහ අවමය wrap කරල තියාගන්නව

♣What is a Wrapper Class?

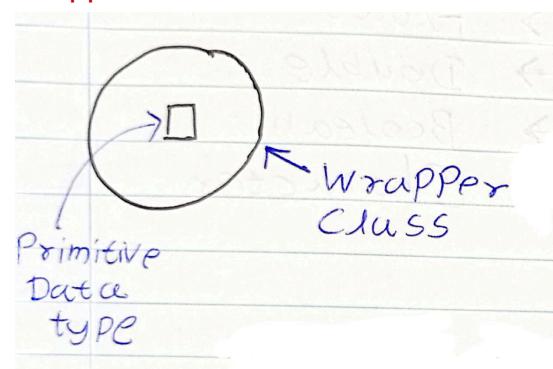
- A wrapper class is like a box, that holds a primitive value and gives it more abilities

Exam answer ,

Wrapper classes in Java are used to convert **primitive data types** into **objects**
convert **primitive data types** into **objects** <-mehema liwath hari

(Byte.MAX_VALUE)

|
Wrapper Class



- අපි class එකෙන් තමයි max value එක ඉල්ලන්නේ. ඒක class එකට තියෙන හැකියාවක් (අපට direct ගිහිල්ල primitive data type එකෙන් මේක ඉල්ලන්න බැ) ඇති නොවනු ලබයි
- Wrapper class විෂ්ට්‍රීත ප්‍රිම්ටිච් දාතාත්‍රීය වල උපරිමය සහ අවමය wrap කරල තියාගන්නව
- Wrapper classes provide a way to use primitive data types (int, boolean, etc...) as objects

Primitive Data Type

byte

short

int

long

float

double

boolean

char

Wrapper Class

Byte

Short

Integer

Long

Float

Double

Boolean

Character

Floating Point

Float

```
class Main {  
    public static void main(String[] args) {  
        float f = 10.5f;  
        System.out.println(f);  
  
        System.out.println(Float.MAX_VALUE); // 3.4028235E38  
        System.out.println(Float.MIN_VALUE); // 1.4E-45  
    }  
}
```

10.5
3.4028235E38
1.4E-45

Double

```
class Main {  
    public static void main(String[] args) {  
        double d = 10.0;  
        System.out.println(d);  
  
        System.out.println(Double.MAX_VALUE); // 1.7976931348623157E308  
        System.out.println(Double.MIN_VALUE); // 4.9E-324  
    }  
}
```

10.0
1.7976931348623157E308
4.9E-324

Char

```
class Main {  
    public static void main(String[] args) {  
        char c = 'A';  
        c = 'a';  
        // c = "a"; // illegal  
        // c = 'hello'; // illegal  
        // c = "hello"; // error: incompatible types  
        System.out.println(c);  
    }  
}
```

```
        // char also has a wrapper class named "Character"
    }
}
a
```

Boolean

```
class Main {
    public static void main(String[] args) {
        // true or false
        boolean b = true;
        b = false;
        System.out.println(b);
    }
}
false
```

```
class Main {
    public static void main(String[] args) {
        // true & false
        boolean b = true;
        b = false;
        System.out.println(b); //false
        // b = "true"; // illegal - incompatible types
        b = 3 > 2; //true , 2 > 3 ? false
        System.out.println(b);
    }
}
false
true
```

● Non-Primitive Data Types

Simple definition for the Non-primitive data types,

Non-Primitive Data Types

Types that are made from primitive types and can store multiple values or have more complex structures, like objects, strings, and arrays.



- Complex multi value data type
- Non-primitive data types are complex data types that can store multiple values or a combination of different values, such as arrays, objects, and classes.
- Non-primitive data types are the building blocks used to create complex structures that can store multiple or grouped values in a program.

String

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

```
    String s = "Hello World";  
  
    System.out.println(s);  
}  
}
```

Hello World

Reserved words



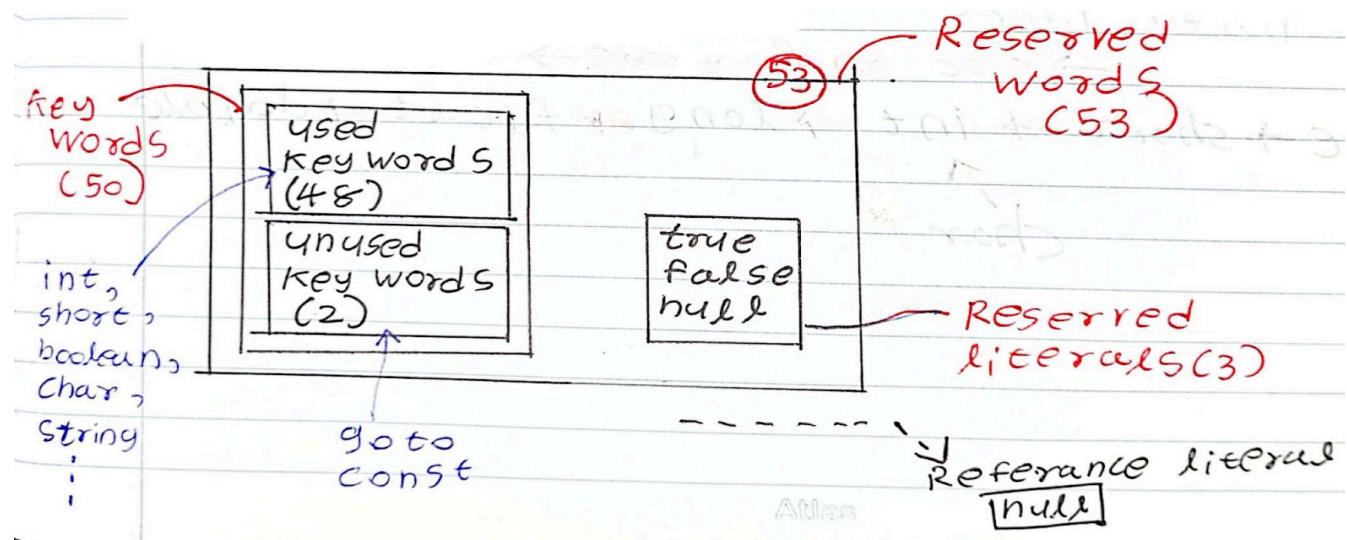
◆ Java Reserved Words

Java reserved words are predefined words with special meanings to the compiler.

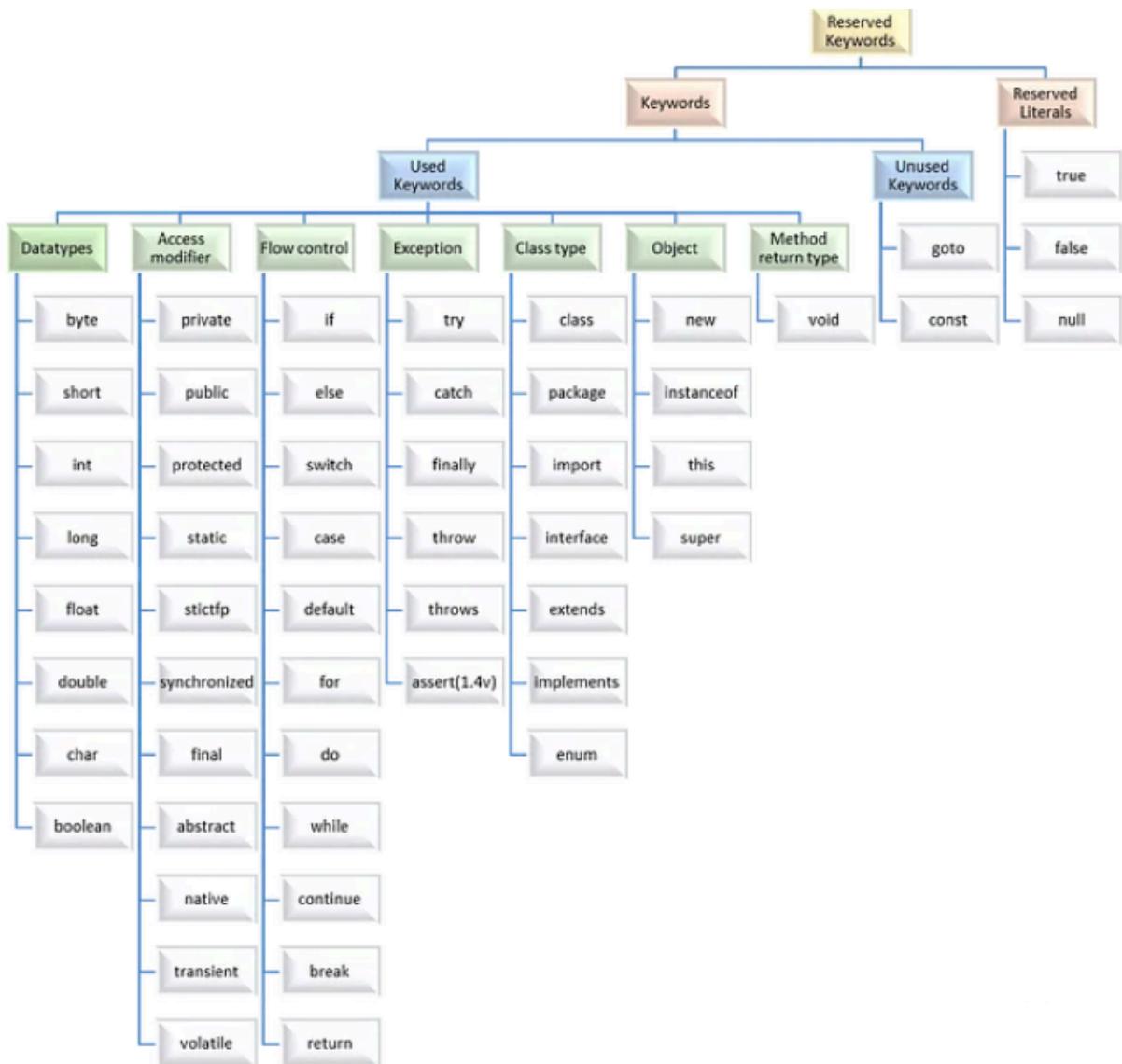
53

-	catch	double	float	int	private	super	true
abstract	char	else	for	interface	protected	switch	try
assert	class	enum	goto	long	public	synchronized	void
boolean	const	extends	if	native	return	this	volatile
break	continue	false	implements	new	short	throw	while
byte	default	final	import	null	static	throws	
case	do	finally	instanceof	package	strictfp	transient	

- Java ඇතුළේ කළින්ම විශේෂ තේරුමක් තියෙන words.
- Compiler තමයි මේව දන්නේ.



Methana string kiyana eka key word ekak nemei
E wagema main , length , Random kiyana ewath keywords nemei



◆ Identifier

```

Class Example{
  Public static void main(String args[]);
  Int x = 2;
}
  
```

Example - class ekata dunnu nama [it is Identifiers] (can change)
 (.java file eke namama thamai api normally mekata denne)

main - method name (can't change) [Identifiers]

args - (can change)

x - Variable name (can't change)

Rules of Variable Name

1. A variable name can consist of Capital letters A-Z, lowercase letters a-z digits 0-9, and two special characters such as _ underscore and \$ dollar sign.
2. The first character must not be a digit.
3. Blank spaces cannot be used in variable names.
4. Variable names are case-sensitive.
5. Java keywords cannot be used as variable names.



```
class Main {  
    public static void main(String[] args) {  
        int mynumber = 10;  
        // int my number = 10; // space thiyanne ba variable ala  
        int num1 = 10;  
        // int lnum = 10; // variable name eka issarahata number ekak danna ba  
        int _myNumber = 10;  
        int $myNumber = 10;  
        // int short = 10; // variable name ekakata reserved Word (keyword)  
ekak danna ba  
        // int void = 10; // variable name ekakata keyword ekak danna ba  
    }  
}
```

```
class Main {  
    public static void main(String[] args) {  
        int mynumber = 10;  
        // int my number = 10;  
        int num1 = 10;  
        // int lnum = 10;  
        int _myNumber = 10;
```

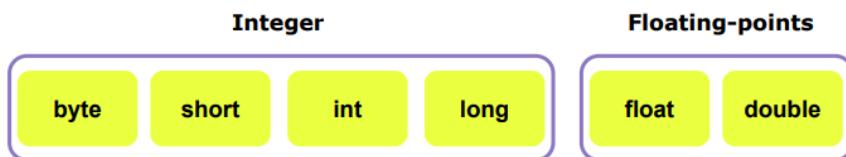
```
        int $myNumber = 10;  
    //    int short = 10;  
    //    int void = 10;  
}  
}
```

```
class Main {  
    public static void main(String[] args) {  
        int mynumber = 10; // valid  
        // int my number = 10; // invalid  
        int num1 = 10; // valid  
        // int lnum = 10; // invalid  
        int _myNumber = 10; // valid  
        int $myNumber = 10; // valid  
        // int short = 10; // invalid  
        // int void = 10; // invalid  
    }  
}
```

◆ Conversion & Casting

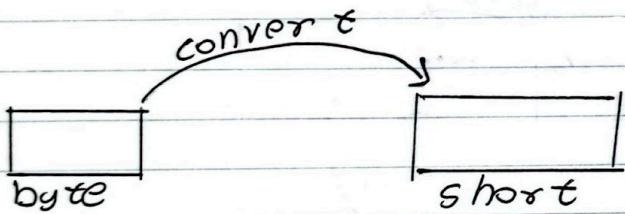
Numeric Data Types

Numeric primitive data types are used to store numbers.



විත Data type තුනක් සහ මිනින්ද නිර්මාණ කළ මෙන් Data type එකට Convert තැබූ ඇත
conversion and Casting

e.g.



Conversion,

- Convert ගැනීමෙහි Programmer involve ගැනීම පෙන්වනු ලබයි
Java Automatically ගැනීම මා උග්‍රණ නිස්සු නිස්සු

Casting,

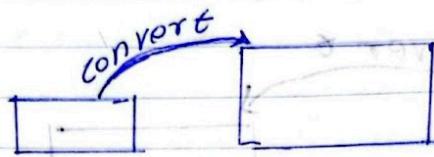
- Convert තැබූ මෙයි Developer මැඳුන්වා මෙයි නියම නියම නියම නියම නියම

□ ഒരു തിരിച്ചറയ് എന്നതാണ് 4009 620

Widening Conversion

Widening Casting

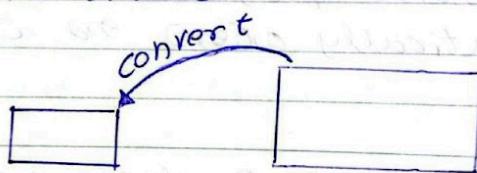
size കുറഞ്ഞ സെക്കി Data type അല്ലെങ്കിൽ Data type കൂടുതൽ convert നൽകാം.



Narrowing Conversion

Narrowing Casting

size കുറഞ്ഞ ലോറു Data type അല്ലെങ്കിൽ Data type കൂടുതൽ convert നൽകാം



Type conversion

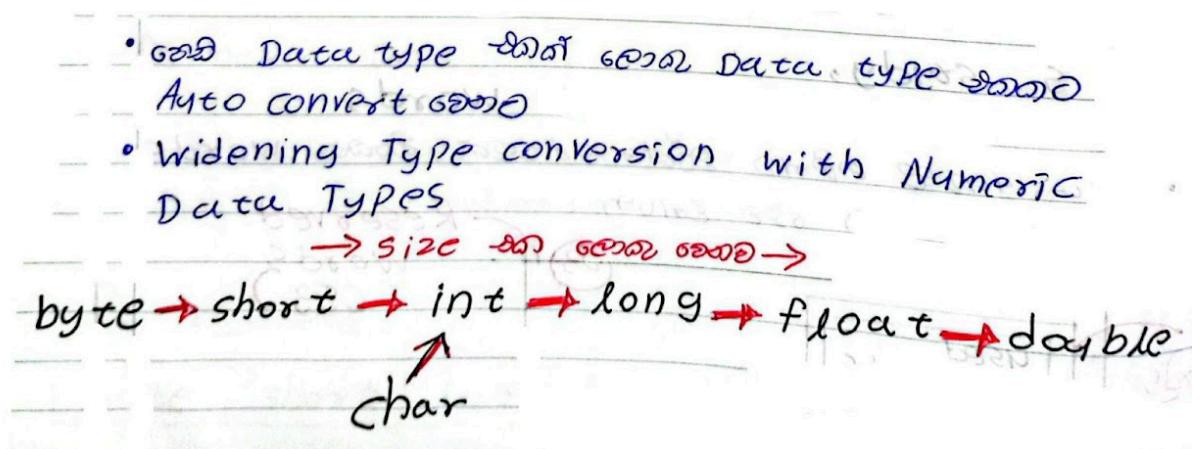
- Widening conversion
- Narrowing conversion

Type Casting

- Widening Casting
- Narrowing Casting

● Widening Conversion

පොඩී data type එකක් ලෙසු data type එකකට Auto Convert වෙනවා



Widening Type Conversion with Numeric Data Types

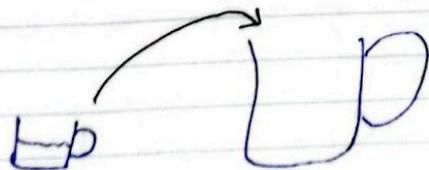


→ size එක ලෙසු වෙනවා →

```
class Main {  
    public static void main(String[] args) {  
        byte b = 10; // byte  
        short s = b; // widening conversion  
        System.out.println(s);  
        int i = s; // widening conversion  
        System.out.println(i);  
        long l = i; // widening conversion  
        System.out.println(l);  
        float f = l; // widening conversion  
        System.out.println(f);  
        double d = f; // widening conversion  
        System.out.println(d);  
    }  
}
```

10
10
10.0
10.0

Widening Conversion real world example,



- easy,
- ക്രോസിംഗ് സ്റ്റേറ്റിംഗ് ആണ്
- മിനാർ ഫാൻസി ഫോറോം ആണ്
- ശിഖാദി ദാഖല ഫോറോം

```
class Main {  
    public static void main(String[] args) {  
        byte b = 10; // byte  
        short s = b; //short, widening Conversion  
  
        System.out.println(s);  
  
        int i = b; //int, widening Conversion  
  
        System.out.println(i);  
    }  
}
```

10
10

```
class Main {  
    public static void main(String[] args) {  
        byte b = 10; // byte  
        short s = b; //short, Widening Conversion  
  
        System.out.println(s);  
  
        int i = b;  
  
        System.out.println(i); //int, Widening Conversion  
  
        long l = i;  
  
        System.out.println(i); //long, Widening Conversion  
    }  
}
```

10
10
10

```
class Main {  
    public static void main(String[] args) {  
        byte b = 10; // byte  
        short s = b; // widening conversion  
        System.out.println(s);  
        int i = s; // widening conversion  
        System.out.println(i);  
        long l = i; // widening conversion  
        System.out.println(l);  
        float f = l; // widening conversion  
        System.out.println(f);  
        double d = f; // widening conversion  
        System.out.println(d);  
    }  
}
```

10
10
10
10.0
10.0

```
class Main {  
    public static void main(String[] args) {  
        byte x = 10; // byte  
        short z = x; // widening conversion  
        int y = x; // widening conversion  
        long l = x; // widening conversion  
        float f = x; // widening conversion  
        double d = x; // widening conversion  
  
        System.out.println(x);  
        System.out.println(z);  
        System.out.println(y);  
        System.out.println(l);  
        System.out.println(f);  
        System.out.println(d);  
    }  
}
```

```
}
```

```
10
```

```
10
```

```
10
```

```
10
```

```
10.0
```

```
10.0
```

```
class Main {  
    public static void main(String[] args) {  
        byte x = 10; // byte  
        short s = x; // widening conversion  
        int i = s; // widening conversion  
        long l = i; // widening conversion  
        float f = l; // widening conversion  
        double d = f; // widening conversion  
  
        System.out.println(d);  
    }  
}
```

```
10.0
```

● Narrowing Casting

ලොකු data type එකක් පොඩි data type එකකට Manual Convert වෙනව

Narrowing Type Conversion with Numeric Data Types

byte ← short ← int ← long ← float ← double

→ size එක ලොකු වෙනව →

byte b = (byte)s;

Casting Operator

```
class Main {  
    public static void main(String[] args) {  
        short s = 10;  
        // byte b = s; // error: incompatible types  
        byte b = (byte)s; // 10  
  
        System.out.println(b);  
    }  
}
```

😢 data lost

```
class Main {  
    public static void main(String[] args) {  
        short s = 128;  
        // byte b = s; // error: incompatible types  
        byte b = (byte)s; // -128  
  
        System.out.println(b);  
    }  
}
```

😢 data lost

```
class Main {
    public static void main(String[] args) {
        short s = 32767;
        // byte b = s; // error: incompatible types
        byte b = (byte)s; // -123

        System.out.println(b);
    }
}
```

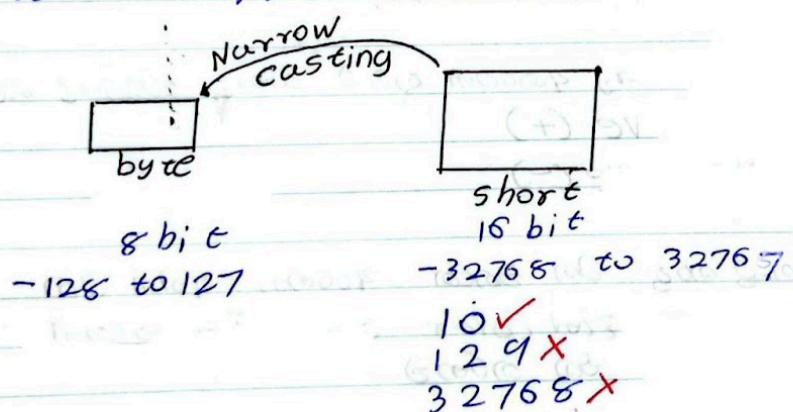
-123

Data lost ekak wenawa harawanna thiyena eke limit eka pannoth

අපේරිත ප්‍රාග ඇඟිල්. Data Lost සිංහ නො
වෙත යොමු කළ ඇති අභ්‍යන්තර ප්‍රාග ඇඟිල්.

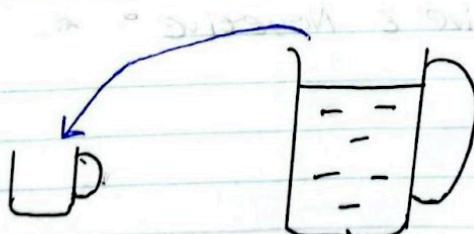
Value එක මෙය data type න්‍යා තුළ ඇති අභ්‍යන්තර
Data type එකට පැමුණු තීවුණු ඇති අභ්‍යන්තර Data
භාෂිත. (Data lost සිංහ වෙත නොවා නොවා
කියනු ඇති limit එක පැමුණු කළ ඇති)

∴ ගොනා ඇති අභ්‍යන්තර නොවා නොවා



Real Word example for Narrow Casting

- complex
- පළු ගෘහනා බා
- තුරර අනුම් යොව
- සිත්සාව දැස්නායුද
(සැවා සිංහ නිකු)



Narrow Casting එහි න්‍යුතුවෙන් සැක්ස්යාංශ ප්‍රතිඵලිය යොමු කළ ඇති අංක ප්‍රතිඵලිය මෙහෙයුම් නොවේ.

Narrow casting and Narrow conversion නොමැත්තුව වේ උග්‍රයෙන් නොවේ.

How to Lost Data

e.g. $\text{short } s = 32767,$

$\text{byte } b = (\text{byte}) s,$

32767 හි Binary තුව ගෙන byte බෙවක සේ

$01111111|11111111$

විලා ඩියෝ binary අගෙන් තුවයි මිනින් පිටුවා පිටුවා නොවනු ලබයා.

1 - Positive (+)

0 - Negative (-)

දැන, ගොඟ රැනු වන මිනින් පිටුවා අභින් වර තිබූ තිබූ නොව. දැන, 101010 තර ගැනීම එකා පිටුවා calculation හි නොවනු ලබයා.

~~01111111|11111111~~

Java, Positive න්‍යුතුව න්‍යුතුව නොවනු ලබයා.

∴ එය Negative (\rightarrow) Number වේ.

$\therefore 1 \rightarrow 0$
 $0 \rightarrow 1$ යනුව ජ්‍යෙ වියා.

$\begin{array}{r} 1111111 \\ \downarrow \\ 0000000 \end{array}$ $\rightarrow 1^{\text{st}} \text{ complement.}$

වෙතෙහි අනින් පැමි තුවකට 1^{st} complement යොහා

වෙත ඇති මූල්‍ය නොවා

$0000000 - 1^{\text{st}} \text{ complement}$

$\begin{array}{r} 0000000 \\ + 1 \\ \hline 0000001 \end{array}$ $\rightarrow 2^{\text{nd}} \text{ complement}$

1^{st} complement නිස් මූල්‍ය තුවකට මැඟි යොහා
 2^{nd} complement යොහා

Result නිස් decimal තුවක

$$\begin{array}{r} +2^2 \times 0 + 2^1 \times 0 + 2^0 \times 1 \\ + 0 + 0 + 1 \times 1 \\ = 1 \end{array}$$

අනු නිගේටී යැයි විලදී සෙයෙහි ගැන වියා

$$= \underline{\underline{-1}}$$

e.g. short s = 32639; \rightarrow byte b = (byte)s;

111111 | 011111
↑
1 නිකුත් Positive (+)

* Positive හෝ 1st 2nd complement නොමැත්තා
හැරු, ගෙවීමට Decimal නො convert කළ මෙයින් නොමැත්තා නොමැත්තා

0111111

$$2^7 \times 0 + 2^6 \times 1 + 2^5 \times 1 + 2^4 \times 1 + 2^3 \times 1 + 2^2 \times 1 + 2^1 \times 1 + 2^0 \times 1$$

$$0 + 64 + 32 + 16 + 8 + 4 + 2 + 1$$

$$96 + 16 + 15$$

$$= +127$$

```
Main {
    public static void main(String[] args) {
        short s = 10;
        // byte b = s; // error: incompatible types
        // byte b = (byte)s; // 10

        System.out.println(b);
    }
}
```

int -> byte

😢 data lost

```
class Main {

    public static void main(String[] args) {
        int i = 2147483647;
        byte b = (byte)i; // ?
    }
}
```

```
        System.out.println(b); //  
    }  
}  
-1
```

```
class Main {  
  
    public static void main(String[] args) {  
  
        long l = 123456;  
  
        // byte b = l; // error: incompatible types  
  
        // byte b = (byte)l;  
  
        byte b = 0;  
  
        b+=l; // narrowing conversion  
  
    }  
}
```

```
class Main {  
    public static void main(String[] args) {  
        int total = 987;  
        int count = 10;  
  
        double avg;  
        avg = total/count; // 98.0  
  
        avg = (double)total/count; // wider casting  
        System.out.println(avg); // 98.7  
  
        avg = total/(double)count; // wider casting  
        System.out.println(avg); // 98.7  
    }  
}  
98.7  
98.7
```

DATA TYPE

• int $x = 20;$
 ↑ ↑
Variable Data
Type Type

+ Data සිංහල Variable එකට assign කිරීම නියම
සංඛ්‍යාව Data Type මඟින් අභ්‍යන්තර ප්‍රාග්ධනය

- Variable එකට assign කරන දත්තවල වර්ගය
- A **data type** defines the **type of data** a variable can hold
- Data type එක කියන්නේ variable එකකු තබාගන්න දත්ත වර්ගය (type of data) දක්වන එකක්.
-  Data type එකෙන් කියන්නේ "මොන වගේ දත්තයක්ද මේක?" කියන එකයි
- What type of data the variable holds

VARIABLE TYPE

- Yam kisi variable ekakat mona wage DATA Type ekakda assign karanne kiyala karanna giyapu welawata
- Variable එක භාවිතා වන්නේ කොහොද, කොහොමද කියන එක
- A **variable type** refers to **how** and **where** a variable is used in a program — its scope, lifetime, and visibility
- Variable type එක කියන්නේ variable එකක් memory එකේ representation එක, scope එක, lifetime එක වගේ technical ලක්ශණ පදනම් කරගෙන panthiකරණය කරන ක්‍රමයකි
- How/where the variable is used in the program

LITERALS

Source code ekak liyana fix value ekak

Source code එක් යොමු fix value නොවන
සංඛ්‍යා දෙවා Literals නොවැනුවන

fix value $\rightarrow 5 + y \times 5$ ✓

DAY 5

```
class Main {  
    public static void main(String[] args) {  
  
        long l = 123456;  
        // byte b = l; // error //narrow  
        byte b = (byte)l ;  
  
        System.out.println(b);  
  
        // byte -> short -> int -> long -> float -> double  
    }  
}  
//64
```

● Narrowing Conversion

Good Data type න්‍යා පොදු Data type තිබා Auto
Convert කෙනෑ

long → byte
class Main {
 public static void main (String args) {
 Long l = 123456;
 // byte b = l; // error: incompatible types
 // byte b = (byte)l; // narrowing casting

byte b=0;

b += 1; // narrowing Convection

$b = b + 1$

System.out.println(b);

}

}

- His byte ekak athulata long ekak daanawa wage wenne,
- Loku kenek (long) podi kenek (byte) widihata podi histhnakata ekathu karanna

```
class Main {  
    public static void main(String[] args) {  
  
        long a = 3;  
        byte b = 2;  
  
        a +=b; //a = a + b  
  
        System.out.println(a);  
  
    }  
}
```

```
class Main {  
    public static void main(String[] args) {  
        long l = 123456;  
        // byte b = l; // error: incompatible types  
  
        // byte b = (byte)l; // narrowing casting  
  
        byte b = 0;  
        b+=1; // (b = b + 1) narrowing convention  
  
        System.out.println(b);  
    }  
}
```

● Widening Casting

കുറി ഡാട്ട ടൈപ് നിന്ന് ലോറ ഡാട്ട ടൈപ് ഫോറ
കുറി കൺവർട്ട് ചെയ്യാൻ പോലീ

int → double

```
class Main {  
    public static void main (String [] args) {  
        int total = 987;  
        int count = 10;  
  
        double avg;  
        avg = total / count; // 987/10 = 98.0?  
        System.out.println (avg);  
    }  
}
```

avg = (double) total / count; // wider casting
System.out.println (avg); // 987/10 = 98.7

avg = total / (double)count; // wider casting
System.out.println (avg); // 987/10 = 98.7

// total, count റേജിംഗേഴ്സ് അവയെല്ലാം double ഫോറ
ചെയ്യാൻ calculate ചെയ്യാം

}

}

ഉള്ളിരുത്തുന്ന ഫോറ int total ആണ്

total ഫോറുന്ന ഡാറ്റ ലോറ കേണ്ടത്
കുറി ശിഖ Widening casting വിന്റെ ഒരു വിഷയമാണ്

```
class Main {  
    public static void main (String [] args) {  
        int total = 987;  
        int count = 10;  
  
        double avg;  
        avg = total / count; // 987/10 = 98.0 ?  
        System.out.println (avg);  
  
        avg = (double) total / count; // wider casting  
        System.out.println (avg); // 987/10 = 98.7  
  
        avg = total / (double) count; // wider casting
```

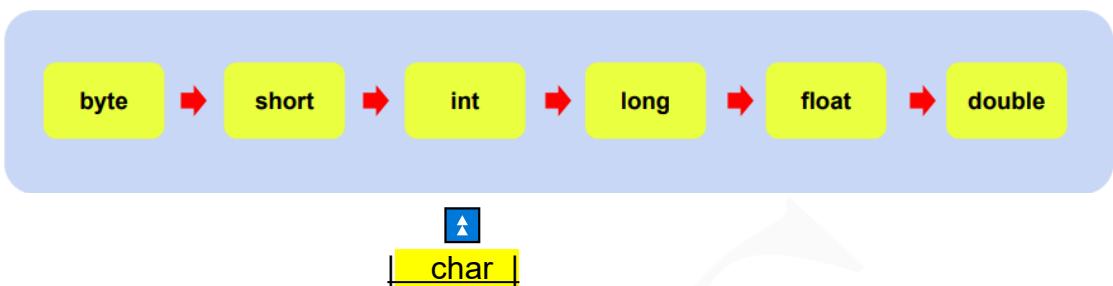
```

        System.out.println(avg); // 987/10 = 98.7

        //me dennagen ekkenek duble karala dunnama haiyata calculate eka
karanawa
    }
}

```

Char



byte → short → int → long → float → double

↑
char ← basic එක
Numeric හිත සහ පැහැදිලිව

char → int (widening conversion)

char වේ අදාළ සටහන වෙත characters

ASCII code එකිනෝ represent ගෙන ඇතුළු

item 000 තියන් යුතු numbers.

class Main {

public static void main (String args[]) {

char c = 'A';

int i = c;

System.out.println(i);

}

//65

↑

A character represent ගෙන ගී 65 මිලිමිටර් මිත්‍රා

Char ->int (widening conversion)

Mekawenne characters ascii code wakin represent wena nisa

```
class Main {  
    public static void main(String[] args) {  
  
        char c = 'A';  
        int i = c;  
  
        System.out.println(i);  
    }  
}
```

//65 (A wala ascii walin represent wnne 65 widihata nisa)

```
class Main {  
    public static void main(String[] args) {  
  
        char c = 65;  
  
        System.out.println(c);  
    }  
}
```

A

◆ Arithmetic Operators

Arithmetic operators in Java are used to perform basic mathematical operations on numeric values (variables or literals).



Basic Arithmetic Operators in maths

$3 + 2$

$3 - 2$

3×2

$3 / 2$

Operator	Name	Description	Example
+	Addition	Adds together two values	$x + y$
-	Subtraction	Subtracts one value from another	$x - y$
*	Multiplication	Multiplies two values	$x * y$
/	Division	Divides one value by another	x / y
%	Modulus	Returns the division remainder	$x \% y$
++	Increment	Increases the value of a variable by 1	$++x$
--	Decrement	Decreases the value of a variable by 1	$--x$

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

        System.out.println(10 + 20); // 30

        System.out.println(20 - 10); // 10

        System.out.println(2 * 3); // 6

        System.out.println(3 / 2); // 1
    }
}
```

```
30
10
6
1
```

```
class Main {
    public static void main(String[] args) {
        System.out.println(10 + 20); // 30
        System.out.println(20 + 10 + 30 + 40); // 100
        System.out.println(20 - 10); // 10
        System.out.println(2 * 3); // 6
        System.out.println(3 / 2); // 1
        System.out.println(6 / 3); // 2
        System.out.println(5 / 2); // 2.5
        System.out.println(5 % 2); // 1
        System.out.println(9 % 2); // 1
        System.out.println(4 % 2); // 0
        System.out.println(9 % 3); // 0
        System.out.println(7 % 4); // 3
    }
}
```

```
30
100
10
6
1
2
2
1
1
0
0
3
```

Variable with arithmetic operators,

```
class Main {  
    public static void main(String[] args) {  
  
        int x = 20;  
        int y = 10;  
  
        System.out.println(x + y); // 20 + 10 = 30  
        System.out.println(x - y); // 20 - 10 = 10  
        System.out.println(x * y); // 20 * 10 = 200  
        System.out.println(x / y); // 20 / 10 = 2  
        System.out.println(x % y); // 20 % 10 = 0  
    }  
}  
30  
10  
200  
2  
0
```

Question: Basic Mathematical Operations in Java

Write a Java program that prompts the user to enter two numbers from the keyboard. After reading the input, perform the following basic mathematical operations and display the results:

1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Modulus (remainder)

Requirements:

- Use the **Scanner** class to read input from the keyboard.
- Handle division carefully to avoid division by zero errors.
- Display the results of each operation clearly.

```
import java.util.Scanner;
class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.println("Enter first number: ");
        int num1 = input.nextInt();

        System.out.println("Enter second number: ");
        int num2 = input.nextInt();

        int add = num1 + num2;
        int sub = num1 - num2;
        int mul = num1 * num2;
        int div = num1 / num2;
        int mod = num1 % num2;

        System.out.println("num1 + num2 = " + add);
        System.out.println("num1 - num2 = " + sub);
        System.out.println("num1 * num2 = " + mul);
        System.out.println("num1 / num2 = " + div);
        System.out.println("num1 % num2 = " + mod);
    }
}
```

Enter first number:

4

Enter second number:

5

num1 + num2 = 9

num1 - num2 = -1

num1 * num2 = 20

num1 / num2 = 0

num1 % num2 = 4

```

import java.util.Scanner;
class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.println("Enter first number: ");
        int num1 = input.nextInt();
        System.out.println("Enter second number: ");
        int num2 = input.nextInt();

        int add = num1 + num2;
        int sub = num1 - num2;
        int mul = num1 * num2;
        int div = num1 / num2;
        int mod = num1 % num2;

        System.out.println(num1 + " + " + num2 + " = " + add);
        System.out.println(num1 + " - " + num2 + " = " + sub);
        System.out.println(num1 + " * " + num2 + " = " + mul);
        System.out.println(num1 + " / " + num2 + " = " + div);
        System.out.println(num1 + " % " + num2 + " = " + mod);
    }
}

```

Enter first number:

45

Enter second number:

65

45 + 65 = 110

45 - 65 = -20

45 * 65 = 2925

45 / 65 = 0

45 % 65 = 45

Value 5k input karama sewage ekathuwa ena widihata hadanna

```

import java.util.Scanner;
class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.println("Enter number 1: ");
        int num1 = input.nextInt();
        System.out.println("Enter number 2: ");
        int num2 = input.nextInt();
        System.out.println("Enter number 3: ");
        int num3 = input.nextInt();
        System.out.println("Enter number 4: ");

```

```
int num4 = input.nextInt();
System.out.println("Enter number 5: ");
int num5 = input.nextInt();

int sum = num1 + num2 + num3 + num4 + num5;

System.out.println(num1 + " + " + num2 + " + " + num3 + " + " + num4
+ " + " + num5 + " = " + sum);

    }
}
```

Enter number 1:

1

Enter number 2:

2

Enter number 3:

3

Enter number 4:

4

Enter number 5:

5

$1 + 2 + 3 + 4 + 5 = 15$

```
import java.util.Scanner;
class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.println("Enter number 1: ");
        int num1 = input.nextInt();
        System.out.println("Enter number 2: ");
        int num2 = input.nextInt();
        System.out.println("Enter number 3: ");
        int num3 = input.nextInt();
        System.out.println("Enter number 4: ");
        int num4 = input.nextInt();
        System.out.println("Enter number 5: ");
        int num5 = input.nextInt();

        System.out.println(num1 + " + " + num2 + " + " + num3 + " + " + num4
+ " + " + num5 + " = " + (num1 + num2 + num3 + num4 + num5));
```

```
    }  
}
```

Enter number 1:

1

Enter number 2:

2

Enter number 3:

3

Enter number 4:

4

Enter number 5:

5

$$1 + 2 + 3 + 4 + 5 = 15$$

◆ Pre-Condition and Post-Condition

Pre-condition

First increments/decrements
the variable,



Then returns the original value.

$++x$ $--x$

Post-condition

First returns the original value,



Then increments/decrements
the variable.

$x++$ $x--$

● Pre Increment (++x)

```
class Main {  
    public static void main(String[] args) {  
        int x = 100;  
        System.out.println(x); // 100  
  
        ++x; // x = x + 1  
        System.out.println(x); // 101  
  
        ++x; // x = x + 1  
        System.out.println(x); // 102  
    }  
}  
100  
101  
102  
]  

```

● Pre Decrement (--x)

```
class Main {  
    public static void main(String[] args) {  
        int x = 100;  
        System.out.println(x); // 100  
  
        --x; // x = x - 1  
        System.out.println(x); // 99  
  
        --x; // x = x - 1  
        System.out.println(x); // 98  
    }  
}  
100  
99  
98
```

● Post increment (x++)

```
class Main {  
    public static void main(String[] args) {  
        int x = 100;  
        System.out.println(x); // 100  
  
        x++; // x = x + 1  
        System.out.println(x); // 101  
  
        x++; // x = x + 1  
        System.out.println(x); // 102  
  
    }  
}
```

100

101

102

● Post decrement (x--)

```
class Main {  
    public static void main(String[] args) {  
        int x = 100;  
        System.out.println(x); // 100  
  
        x--; // x = x - 1  
        System.out.println(x); // 99  
  
        x--; // x = x - 1  
        System.out.println(x); // 98  
  
    }  
}
```

100

99

98

```

class Main {
    public static void main(String[] args) {
        int x = 100;
        System.out.println(x); // 100

        x++; // x = x + 1
        System.out.println(x); // 101

        ++x; // x = x + 1
        System.out.println(x); // 102
    }
}

```

100
101
102

```

class Main {
    public static void main(String[] args) {
        int x = 100;
        int y = ++x; // ++x -> x = x + 1 = 101
        System.out.println(y); // 101
    }
}

```

101

```

class Main {
    public static void main(String[] args) {
        int x = 100;
        int y = x++;
        System.out.println(y); // 100
    }
}

```

Value ekak me widihata increment decrement karanna ba

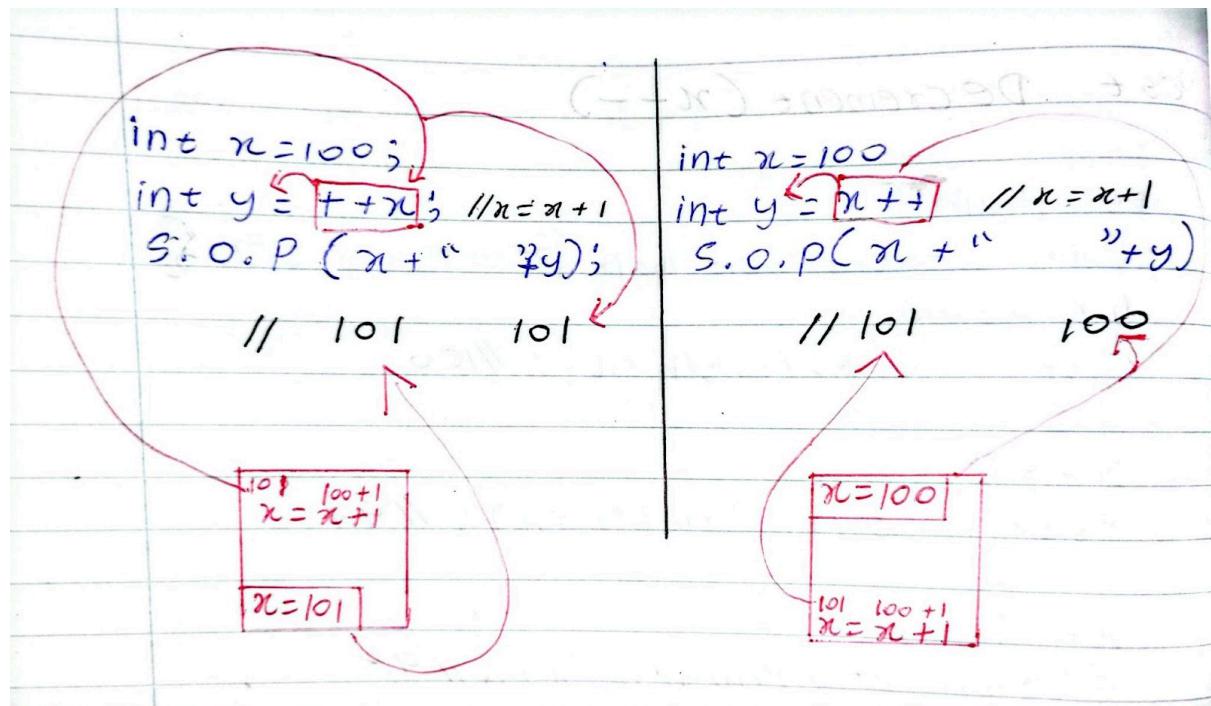
```
class Main{
```

```

public static void main(String[] args) {

    int x = 100;
    // int y = ++10 ; //Error
    System.out.println(y);
}

```



```

class Main {
    public static void main(String[] args) {
        int x = 100;
        int y = ++x; // ++x -> x = x + 1 = 101
        System.out.println(x + " " + y); // 101 101
    }
}

```

101 101

```

class Main {
    public static void main(String[] args) {
        int x = 100;
        int y = x++; // ++x -> x = x + 1 = 101
    }
}

```

```
        System.out.println(x + " " + y); // 101 100
    }
}

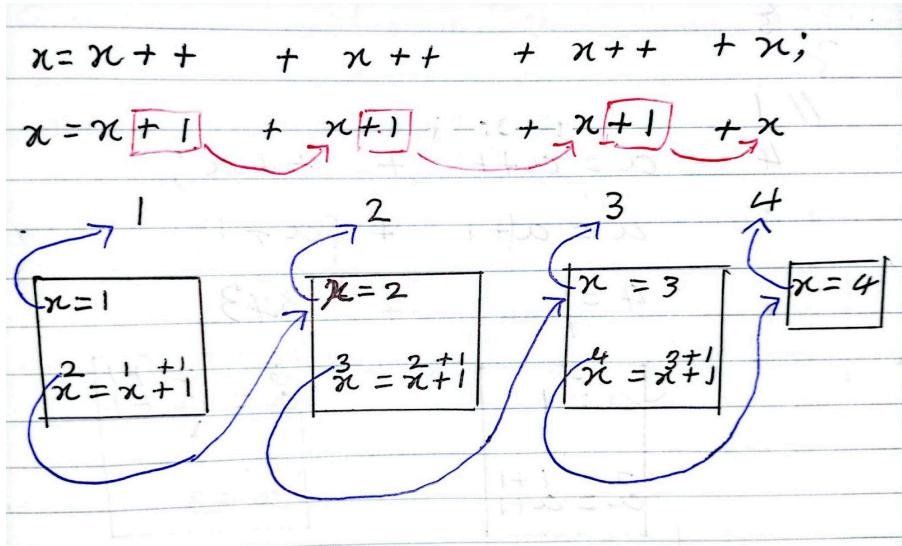
101 100
```

```
class Main {
    public static void main(String[] args) {
        int a = 0;
        System.out.println(a); // Line 01
        a=a++;
        System.out.println(a); // Line 02
        a=++a;
        System.out.println(a); // Line 03
    }
}

0
0
1
```

```
class Main {
    public static void main(String[] args) {
        int a = 1;
        System.out.println(a); // Line 01
        a=a++;
        System.out.println(a); // Line 02
        a=++a;
        System.out.println(a); // Line 03
    }
}

1
1
2
```

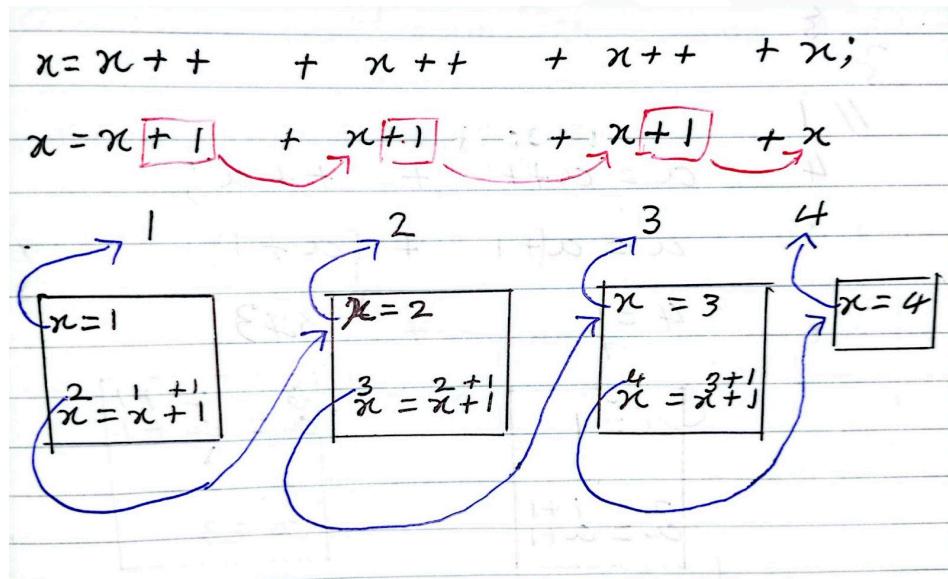


```

class Main {
    public static void main(String[] args) {
        int x = 1;
        x = x++ + x++ + x++ + x;
        System.out.println(x); // x = 1 + 2 + 3 + 4 = 10
    }
}

```

10

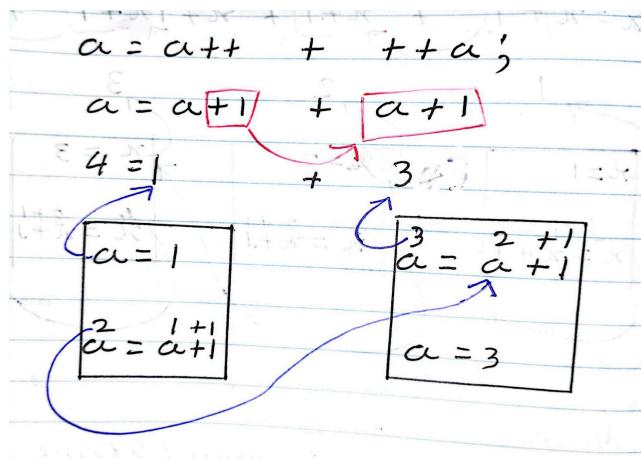


```

class Main {
    public static void main(String[] args) {
        int x = 1;
        x = ++x + ++x + ++x + x;
        System.out.println(x); // x = 2 + 3 + 4 + 4 = 13
    }
}

```

13



```
class Main {
    public static void main(String[] args) {
        int a = 1;
        System.out.println(a); // 1

        a=a++ + ++a;
        System.out.println(a); // 1 + 3 = 4
    }
}
1
4
```

HW

```
class Main {
    public static void main(String[] args) {
        int a = 1, b = 2, c = 3, d = 4;
        int x;
        x = a++ + b++ + c++ + b++;
        System.out.println(a + " " + b + " " + c + " " + d + " " + x); // ?
    }
}
2 4 4 4 9
```

HW

```
class Main {
    public static void main(String[] args) {
        int a=1,b=2,c=3,d=4;
        int x;
        x = ++a + ++b + ++c + ++b;
        System.out.println(a + " " + b + " " + c + " " + d + " " + x); // ?
    }
}
2 4 4 4 13
```

HW

```
class Main {  
    public static void main(String[] args) {  
        int a=1,b=2,c=3,d=4;  
        int x;  
        x = ++a + ++b + ++c - ++b;  
        System.out.println(a + " " + b + " " + c + " " + d + " " + x); // ?  
    }  
}  
2 4 4 4 5
```

HW

```
class Main {  
    public static void main(String[] args) {  
        int a=1,b=2,c=3,d=4;  
        int x;  
        x = ++a - ++b - ++c - ++b;  
        System.out.println(a + " " + b + " " + c + " " + d + " " + x); // ?  
    }  
}  
2 4 4 4 -9
```

```
class Main {  
    public static void main(String[] args) {  
        int x; // define a variable  
        x = 100; // variable initialization  
        int y = 200; // variable defining + variable initialization  
    }  
}
```

Highest precedence

<https://www.geeksforgeeks.org/java/operator-precedence-and-associativity-in-java/>

Operators	Associativity	Type
++ --	Right to left	Unary postfix
++ -- + - ~ ! (type)	Right to left	Unary prefix
* / %	Left to right	Multiplicative
+ -	Left to right	Additive
<< >> >>>	Left to right	Shift
< <= > >=	Left to right	Relational
== !=	Left to right	Equality
&	Left to right	Boolean Logical AND
^	Left to right	Boolean Logical Exclusive OR
	Left to right	Boolean Logical Inclusive OR
&&	Left to right	Conditional AND
	Left to right	Conditional OR
:?	Right to left	Conditional
= += -= *= /= %=	Right to left	Assignment

◆final

- Variable type ekak mulata final danmoth ema variable type eka yata the dapu variable name ekakata initialize karanna puluhan eka paarak witharai
- WENAS NOWANA AWASTHA API USE KARANNE FINAL DAALA

```
class Main {
    public static void main(String[] args) {
        /* final int x; // defining a variable
        x = 100; // variable initialization
        x = 200; // illegal */
        final int x = 200;
        System.out.println(x);
```

```
}
```

200

```
class Main {
    public static void main(String[] args) {
        /* final int x; // variable defining
        x = 100; // variable initialization
        x = 200; // error */
        final int y = 400;
        // y = 500; // error: cannot assign a value to final variable
        System.out.println(y);
    }
}
```

400

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

        final int x = 200 , y;
        y=100;
        // x = 100; // error: cannot assign a another value to
final variable x
        // y = 500; // error: cannot assign a another value to
final variable y
        System.out.println(x);
        System.out.println(y);
    }
}
```

200

100

◆ Compound assignment

Compound assignment
operators, not basic arithmetic
operators.

They combine an arithmetic
operation (+ or -) with
assignment (=).

operator	example	equivalent to
<code>+=</code>	<code>a += b</code>	<code>a = a + b</code>
<code>-=</code>	<code>a -= b</code>	<code>a = a - b</code>
<code>*=</code>	<code>a *= b</code>	<code>a = a * b</code>
<code>/=</code>	<code>a /= b</code>	<code>a = a / b</code>

```
|%=      a % = b      a = a % b |
```

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

        int a = 20;
        int b = 10;

        // a = a + b; // 20 + 10 = 30

        a+=b; // a = a + b

        System.out.println(a); // 30
    }
}
```

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

        int a = 99;
        int b = 10;
        // a+=b; // a = a + b
        // a-=b; // a = a - b
        // a*=b; // a = a * b
        // a/=b; // a = a / b
        a%=b; // a = a % b

        System.out.println(a); // 9
    }
}
```

9

```
class Main {
    public static void main(String[] args) {
        int a = 20;
        int b = 10;
        // a+=b; // a = a + b -> 30
        // a-=b; // a = a - b -> 10
        // a*=b; // a = a * b -> 200
        // a/=b; // a = a / b -> 2
        a%=b; // a = a % b -> 0
        System.out.println(a); // ?
    }
}
```

0

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

        int a = 99;
        int b = 10;
        a+=b; // a = a + b
        System.out.println(a);

        int c = 99;
        int d = 10;
        c-=d; // c = c - d
        System.out.println(c);

        int e = 99;
        int f = 10;
        e*=f; // e = e * f
        System.out.println(e);

        int g = 99;
        int h = 10;
        g/=h; // g = g / h
        System.out.println(g);

        int i = 99;
        int j = 10;
        i%=j; // i = i % j
        System.out.println(i);
    }
}
```

◆ Relation Operators

Relational operators in Java are used to compare two values and return a boolean result (true or false).

Operator	Meaning
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to
==	Equal to
!=	Not equal to

true & false out put dena operator

- $(x > y)$ x wishaalada y ta wada
- $(x < y)$ x kudaada y ta wada
- $(x \geq y)$ x wishaala ho samaanada y ta
- $(x \leq y)$ x kuda ho samaanada y ta
- $(x == y)$ x samaanada y ta
- $(x != y)$ x asamaanada y ta

```
class Main {  
    public static void main(String[] args) {  
  
        System.out.println(20>10); // true  
        System.out.println(20<10); // false  
  
        int x = 100;  
        System.out.println(x>10); // 100>10 ? -> true  
        System.out.println(x<10); // 100<10 ? -> false  
  
        int a = 50;  
        int b = 20;  
        System.out.println(a>b); // 50 > 20 ? ->true  
        System.out.println(a<b); // 50 < 20 ? ->false  
    }  
}
```

true
false
true
false
true
false

```
class Main {  
    public static void main(String[] args) {  
  
        int a = 50;  
        int b = 50;
```

```
        System.out.println(a>b); // 50 > 50 ? ->false
        System.out.println(a<b); // 50 < 50 ? ->false

        System.out.println(a>=b); // 50 >= 50 ? ->true
        System.out.println(a<=b); // 50 <= 50 ? ->true
    }
}

false
false
true
true
```

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

        int a = 50;
        int b = 50;
        int c = 20;

        System.out.println(a == b); // 50 == 50 ? -> true
        System.out.println(a == c); // 50 == 20 ? -> false
    }
}

true
false
```

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

        int a = 50;
        int b = 50;
        int c = 20;

        System.out.println(a != b); // 50 Not Equal 50 ? -> false
        System.out.println(a != c); // 50 Not Equal 20 ? -> true
    }
}

false
true
```

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

        System.out.println(20>10); // true
        System.out.println(20<10); // false

        int x = 100;
        System.out.println(x>10); // 100>10 ? -> true
        System.out.println(x<10); // 100<10 ? -> false
    }
}
```

```
int a = 50;
int b = 20;
int c = 50;
System.out.println(a>b); // 50 > 20 ? ->true
System.out.println(a<b); // 50 < 20 ? ->false

System.out.println(a>c); // 50 > 50 ? ->false
System.out.println(a<c); // 50 < 50 ? ->false

System.out.println(a>=c); // 50 >= 50 ? ->true
System.out.println(a<=c); // 50 <= 50 ? ->true

System.out.println(a == c); // 50 == 50 ? -> true
System.out.println(a == b); // 50 == 20 ? -> false

System.out.println(a != c); // 50 Not Equal 50 ? -> false
System.out.println(a != b); // 50 Not Equal 20 ? -> true
}

}
```

◆ Flow Charts

Symbol	Name	Function
	Start/end	An oval represents a start or end point
	Arrows	A line is a connector that shows relationships between the representative shapes
	Input/Output	A parallelogram represents input or output
	Process	A rectangle represents a process
	Decision	A diamond indicates a decision

Java Flow Controllers

Conditions

if statement, if-else statement,
else-if ladder, switch-case

Loops

for loops, while loops, do-while
loops

Jump

statements

break, continue

Condition

- condition ekata danna puluwan true false out put ena Relational operators witharai

◆ if Statement

```
if (condition) {  
    // executes if condition is true  
}
```

- condition ekata danna puluwan Relational operators witharai
- Api methanadi balaporoththu wenne ema if statement eka hari nam eya yatathe thiyyena de karannath waradinam ethanin if statement eka yatathe thiyyena de nokarannath yana kriyawaliya.

```
class Main {  
    public static void main(String[] args) {  
        if(50>10) { // true  
            System.out.println("Good");  
        }  
    }  
}
```

```
class Main {  
    public static void main(String[] args) {  
        if(50>100) { // false  
            System.out.println("Hello!");  
        }  
    }  
}
```

```
}
```

```
class Main {
    public static void main(String[] args) {
        int x = 30;
        if(x>20) { // 30>20 -> true
            System.out.println("Hello!");
        }
    }
}
```

Hello!

```
class Main {
    public static void main(String[] args) {
        int x = 10;
        if(x>20) { // 10>20 -> false
            System.out.println("Hello!");
        }
    }
}
```

Empty (mean, condition eka false wela thinawa eka nisa mukuth karanne na)

```
class Main {
    public static void main(String[] args) {
        int x = 100;
        if(x>20) {
            System.out.println("Hello 1");
            System.out.println("Hello 2");
            System.out.println("Hello 3");
            System.out.println("Hello 4");
            System.out.println("Hello 5");
        }
    }
}
```

Hello 1

Hello 2

Hello 3

Hello 4

Hello 5

```
class Main {  
    public static void main(String[] args) {  
        int x = 100;  
        if(x>20) {  
            int k = x + 100;  
            System.out.println(k);  
        }  
    }  
}
```

200

summery

```
class Main {  
    public static void main(String[] args) {  
        if(50>10) { // true  
            System.out.println("HELLO 1");  
        }  
        if(50>100) { // false  
            System.out.println("HELLO 2");  
        }  
  
        int x = 30;  
        if(x>20) { // 30>20 -> true  
            System.out.println("HELLO 3");  
            System.out.println("HELLO 4");  
        }  
  
        int y = 10;  
        if(y>20) { // 10>20 -> false  
            System.out.println("HELLO 5");  
        }  
  
        int z = 100;  
        if(x>20) {  
            int k = z + 100;  
            System.out.println(k);  
        }  
    }  
}
```

Marks 70ta wadi nam iphone 16 phone ekak buy karala denawa

```
import java.util.Scanner;
class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.println("Enter your marks : ");
        int marks = input.nextInt();

        if (marks>70) {

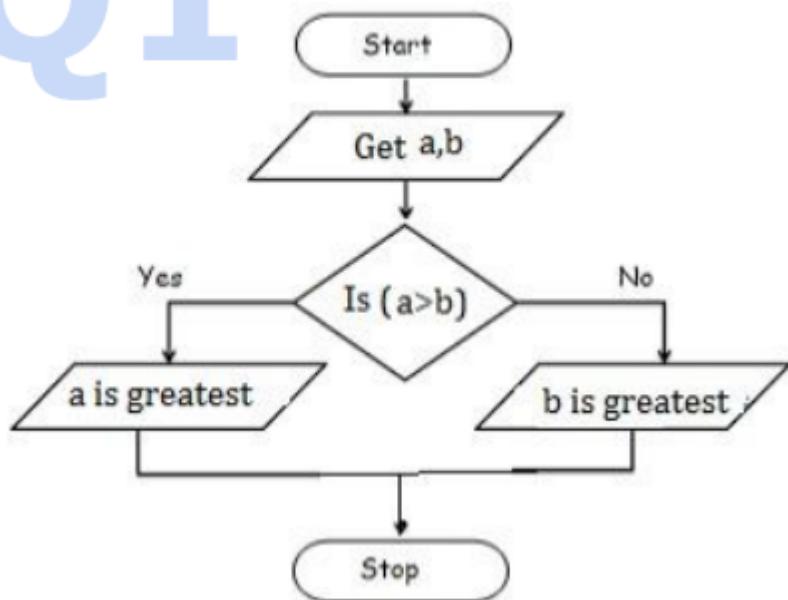
            System.out.println("Let's buy an iphone 16");
        }
    }
}
```

Enter your marks :

87

Let's buy an iphone 16

Q1



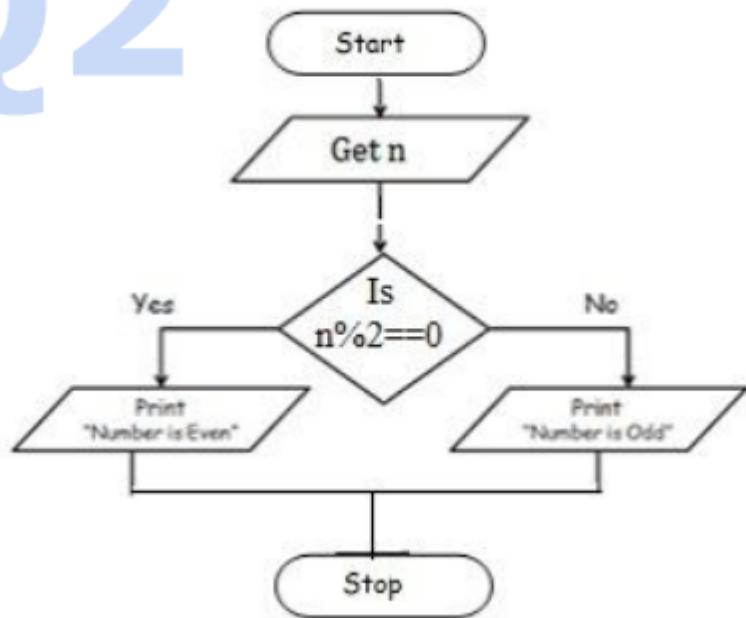
```
import java.util.Scanner;
class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter your value 1: ");
        int a = input.nextInt();

        System.out.print("Enter your value 2: ");
        int b = input.nextInt();

        if (a>b) {
            System.out.println(a + " is greatest");
        }else {
            System.out.println(b + " is greatest");
        }
    }
}
```

Q2



```
import java.util.Scanner;
class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

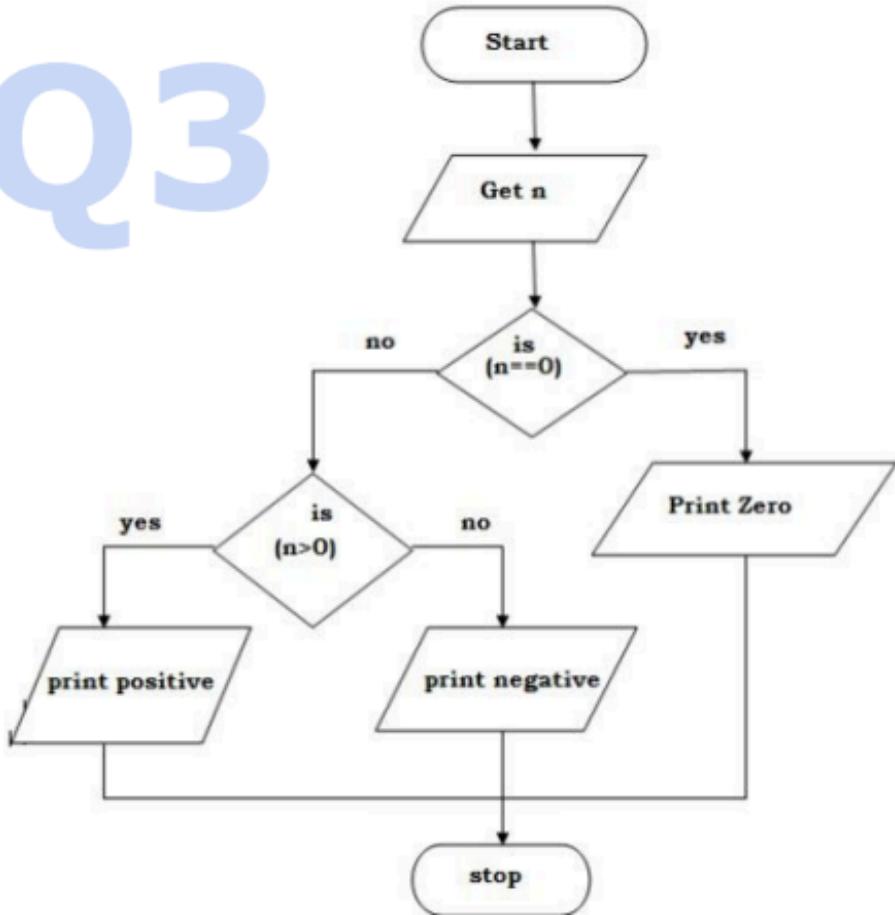
        System.out.print("Enter your number: ");
        int n = input.nextInt();

        if (n % 2 == 0) {
            System.out.println("Number is even");
        } else {
            System.out.println("Number is odd");
        }
    }
}
```

Enter your number: 23

Number is odd

Q3



```
import java.util.Scanner;
class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

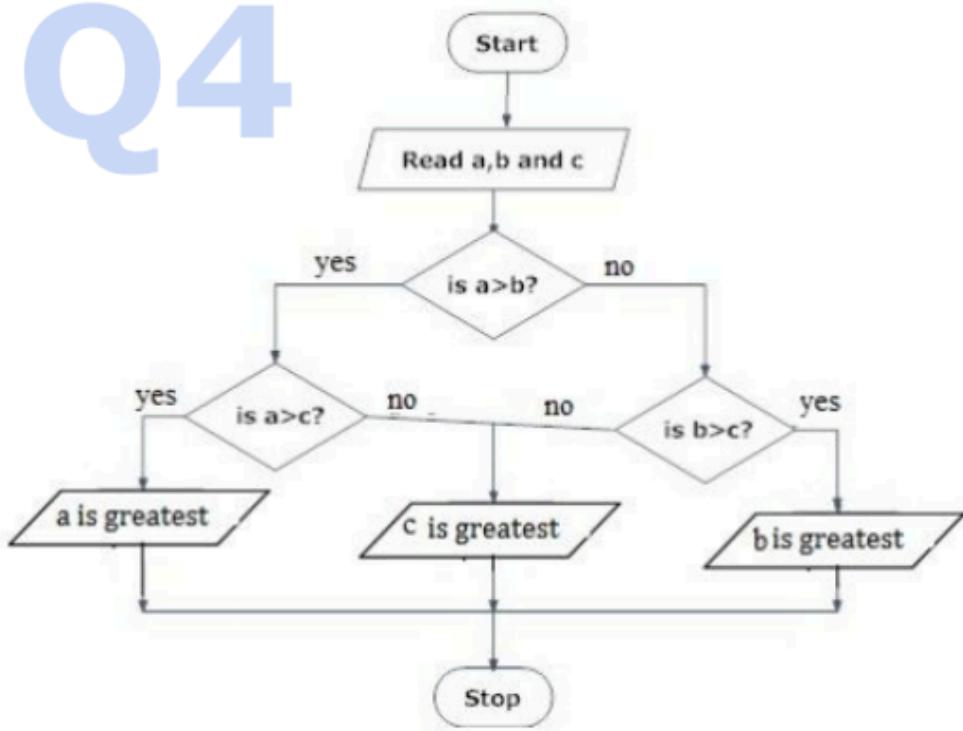
        System.out.print("Enter your value : ");
        int n = input.nextInt();

        if (n == 0) {
            System.out.println("0");
        } else {
            if (n > 0){
                System.out.println("Positive");
            } else {
                System.out.println("Negative");
            }
        }
    }
}
```

Enter your value : -4

Negative

Q4



```
import java.util.Scanner;
class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter your value 1: ");
        int a = input.nextInt();

        System.out.print("Enter your value 2: ");
        int b = input.nextInt();

        System.out.print("Enter your value 3: ");
        int c = input.nextInt();

        if (a > b) {
            if (a>c) {
                System.out.println(a + " is greatest");
            }else {
                System.out.println(c + " is greatest");
            }
        }
    }
}
```

```

}else {
    if (b>c) {
        System.out.println(b + " is greatest");
    }else {
        System.out.println(c + " is greatest");
    }
}

}

```

Enter your value 1: 54

Enter your value 2: 645

Enter your value 3: 35

645 is greatest

OR

```

import java.util.Scanner;

class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter the first number: ");
        int a = input.nextInt();

        System.out.print("Enter the second number: ");
        int b = input.nextInt();

        System.out.print("Enter the third number: ");
        int c = input.nextInt();

        int largest = a;

        if (b > largest) {
            largest = b;
        }
        if (c > largest) {
            largest = c;
        }

        System.out.println("The largest number is: " + largest);
    }
}

```

Enter the first number: 23

Enter the second number: 423

Enter the third number: 123

The largest number is: 423

A university wants to automate its grading system using a Java program.

The grading rules are as follows:

- Grade **A**: Marks ≥ 70
- Grade **B**: Marks ≥ 65 and < 70
- Grade **C**: Marks ≥ 55 and < 65
- Grade **S**: Marks ≥ 45 and < 55
- Grade **F**: Marks < 45

Q5

Tasks:

(i) Draw a flowchart to represent the grading logic.

(ii) Write a Java program that:

- Accepts a student's marks (0–100) as input.
- Validates the input to ensure it's within the correct range.
- Assigns and displays the correct grade based on the grading criteria above.

```
import java.util.Scanner;
class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter your Marks: ");
        int m = input.nextInt();

        if (m >= 0 && m <=100) {
            if (m >= 70) {
                System.out.println("Grade A");
            } else if (m >= 65) {
                System.out.println("Grade B");
            } else if (m >= 55){
                System.out.println("Grade C");
            } else if (m >=45) {
                System.out.println("Grade S");
            }else {
                System.out.println("Grade F");
            }
        }else {
            System.out.println("Invalid Marks");
        }
    }
}
```

```
}
```

Q6

Write a Java program that:

- Accepts **four integer numbers** as input from the user.
- Finds and displays the **largest number** among the four.

```
import java.util.Scanner;

class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter the first number: ");
        int num1 = input.nextInt();

        System.out.print("Enter the second number: ");
        int num2 = input.nextInt();

        System.out.print("Enter the third number: ");
        int num3 = input.nextInt();

        System.out.print("Enter the forth number: ");
        int num4 = input.nextInt();

        int largest = num1;

        if (num2 > largest) {
            largest = num2;
        }
        if (num3 > largest) {
            largest = num3;
        }
        if (num4 > largest) {
            largest = num4;
        }

        System.out.println("The largest number is: " + largest);
    }
}
```

Logical AND operator (`&&`)

`&&` saha yan arutha genadei

```
class Main {  
    public static void main(String args[]) {  
        int m = 50;  
  
        System.out.println(m >= 0 && m <= 100);  
        // මෙහි m 0 - 100 අතර නම් true වේයි  
    }  
}  
true
```

```
class Main {  
    public static void main(String args[]) {  
        int m = 50;  
  
        System.out.println(m >= 0 & m <= 100);  
        // මෙය run වේයි, භැබුදී logic එක bitwise AND එකක්.  
        // නීතියෙන් හර නෑ. වැරදි ලෙස භාවිත කරන එකක්.  
    }  
}  
true
```

 **&&** — නිවැරදි logical AND operator එක (එය භාවිතා කරන්න).

 **&** — bitwise operator එක (logic check වලට භාවිතා නොකරන්න)

DAY - 05

QUIZE

```
int x = 10;
int y = 20;

if (x < 15) {
    if (y > 10) {
        System.out.println("A");
    } else {
        System.out.println("B");
    }
} else {
    System.out.println("C");
}
```

```
int level = 2;

switch (level * 2) {
    case 2:
        System.out.println("Low");
        break;
    case 4:
        System.out.println("Medium");
    case 6:
        System.out.println("High");
        break;
    default:
        System.out.println("Unknown");
}
```

```
int a = 5;
int b = a-- + --a;
System.out.println("a = " + a + ", b = " + b);
```

◆ If-else Statement

```
if (condition) {  
    // executes if condition is true  
}  
else {  
    // executes if condition is false  
}
```

Marks 70ta wadi nam iphone 16 phone ekak buy karala denawa 70ta adu nam Nokia 1100 ekak denawa

```
import java.util.Scanner;  
  
class Main {  
    public static void main(String[] args) {  
        Scanner input = new Scanner(System.in);  
  
        System.out.println("Enter your marks: ");  
        int marks = input.nextInt();  
  
        if(marks>=70) {  
            System.out.println("Let's buy an iphone 16");  
        }  
        if(marks<70) {  
            System.out.println("Let's buy a Nokia 1100");  
        }  
    }  
}
```

Enter your marks:

76

Let's buy an iphone 16

```
class Main {  
    public static void main(String[] args) {  
  
        int marks =80;  
  
        if(marks >= 70) {  
            System.out.println("Let's buy an iphone 16");  
        } else {
```

```
        System.out.println("Let's buy a Nokia 1100");
    }

}

}
```

```
import java.util.Scanner;

class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.println("Enter your marks: ");
        int marks = input.nextInt();

        if(marks >= 70) {
            System.out.println("Let's buy an iphone 16");
        } else {
            System.out.println("Let's buy a Nokia");
        }
    }
}
```

Enter your marks:

89

Let's buy an iphone 16

```
import java.util.Scanner;

class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.println("Enter A: ");
        int a = input.nextInt();

        System.out.println("Enter B: ");
        int b = input.nextInt();

        if(a>b) {
            System.out.println("A is greatest");
        } else {
            System.out.println("B is greatest");
        }
    }
}
```

```
import java.util.Scanner;

class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.println("Enter Number: ");
        int n = input.nextInt();

        if(n % 2 == 0) {
            System.out.println("It is an even number");
        } else {
            System.out.println("It is an odd number");
        }
    }
}
```

```
import java.util.Scanner;

class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.println("Enter Number: ");

        int n = input.nextInt();
        int result = n%2;
        // if that result is zero -> even

        if(result==0) {
            System.out.println("It is an even number");
        } else {
            System.out.println("It is an odd number");
        }
    }
}
```

Number ekak input karala eka zeroda negetiveda positive da kiyala check karanna

```
import java.util.Scanner;
class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.println("Enter Number: ");
        int n = input.nextInt();

        if(n==0) {
            System.out.println("Zero");
        } else {
            if(n>0) {
                System.out.println("Positive Number");
            } else {
                System.out.println("Negetive Number");
            }
        }
    }
}
```

```
import java.util.Scanner;
class Main {
```

```
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter Number: ");
        int n = input.nextInt();

        if(n>0) {
            System.out.println("Positive Number");
        } else {
            System.out.println("Negetive Number");
        }

        System.out.println("Zero");

    }
}
```

//Enter Number: 0
Negative Number
Zero (meka apita ona result eka nemei) me wage else danmata pase thawa ekak liwwoth pahaa ekath liyawenawa

```
import java.util.Scanner;
class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter Number: ");
        int n = input.nextInt();

        if(n>0) {
            System.out.println("Positive Number");
        } else {
            System.out.println("Negetive Number");
        }

        if(n ==0) {
            System.out.println("Zero");
        }
    }
}
```

//Negetive Number

Zero

(meka apita ona result eka nemei)

If else use karoth

```
import java.util.Scanner;
class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter Number: ");
        int n = input.nextInt();

        if(n>0) {
```

```

        System.out.println("Positive Number");
    } else if (n<0) {
        System.out.println("Negetive Number");
    }
else {
    System.out.println("Zero");

}

}
}

```

true kina boolean value eka wachanayak widihata denne mehema, (tenery use karalath me widihata ganna puluwan)

```

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

        boolean oba_harida = true;

        if(oba_harida) {
            System.out.println("Hari ");
        }
        else {
            System.out.println("waradiy ");
        }
    }
}

```

Hari

◆else-if ladder

```

if (1st condition) {
    // executes if 1st condition is true
} else if (2nd condition) {
    // executes if 2nd condition is true
} else {
    // executes if all conditions are false
}

```

Battery Level

```
import java.util.Scanner;
class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter battery Level: ");
        int marks = input.nextInt();

        // > 80-> GOOD, 45-> NORMAL, 20-> LOW, ->DEAD

        if(marks>80) {
            System.out.println("GOOD!");
        } else if(marks>45) {
            System.out.println("NORMAL!");
        } else if (marks>20) {
            System.out.println("LOW!");
        } else {
            System.out.println("DEAD!");
        }
    }
}
```

Enter battery Level: 21

LOW!

```
import java.util.Scanner;
class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.println("Enter Marks: ");
        int marks = input.nextInt();

        // >=70 - A, >=50 - B, F
        if(marks>=70) {
            System.out.println("A");
        } else if(marks>=50) {
            System.out.println("B");
        } else {
            System.out.println("F");
        }
    }
}
```

```
import java.util.Scanner;
class Main {
    public static void main(String[] args) {
```

```

Scanner input = new Scanner(System.in);

System.out.print("Enter Marks: ");
int marks = input.nextInt();

// >=75 - A, >=55 - B, >=45 - C, >=35 - S , F
if(marks>=75) {
    System.out.println("Your Grade is : A");
} else if(marks>=55) {
    System.out.println("Your Grade is : B");
} else if(marks>=45) {
    System.out.println("Your Grade is : C");
} else if(marks>=35) {
    System.out.println("Your Grade is : S");
} else {
    System.out.println("Your Grade is : F");
}

}
}

```

Enter Marks: 57

Your Grade is : B

Marks

```

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

        int m = 678;

        if (m >= 0 && m <=100) {
            if (m >= 70) {
                System.out.println("Grade A");
            } else if (m >= 65) {
                System.out.println("Grade B");
            } else if (m >= 55) {
                System.out.println("Grade C");
            } else if (m >=45) {
                System.out.println("Grade S");
            }else {
                System.out.println("Grade F");
            }
        }else {
            System.out.println("Invalid Marks");
        }
    }
}

```

```
    }
}
```

Optimize widiha

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

        int m = 78;

        if (m >= 75 && m <=100) {
            System.out.println("Grade A");
        } else if (m >= 55 && m <=65) {
            System.out.println("Grade B");
        } else if (m >= 45 && m <=55) {
            System.out.println("Grade C");
        } else if (m >= 35 && m <=45) {
            System.out.println("Grade S");
        }else if (m >= 0 && m <=35) {
            System.out.println("Grade F");
        }else {
            System.out.println("Invalid Marks");
        }
    }
}
```

4 input numbers and give max number

```
import java.util.Scanner;
class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter a: ");
        int a = input.nextInt();

        System.out.print("Enter b: ");
        int b = input.nextInt();
```

```

System.out.print("Enter a: ");
int a = input.nextInt();

System.out.print("Enter b: ");
int b = input.nextInt();

System.out.print("Enter c: ");
int c = input.nextInt();

System.out.print("Enter d: ");
int d = input.nextInt();

int max = a;

if(b > max) {
    max = b;
}
if(c > max) {
    max = c;
}
if(d > max) {
    max = d;
}
System.out.println("max is " + max);

}
}

```

Enter a: 10

Enter b: 20

Enter c: 30

Enter d: 40

max is 40

wrong

```

import java.util.Scanner;
class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter a: ");
        int a = input.nextInt();

        System.out.print("Enter b: ");
        int b = input.nextInt();

        System.out.print("Enter c: ");
        int c = input.nextInt();

        System.out.print("Enter d: ");

```

```

int d = input.nextInt();

int max = a;

if(b > max) {
    max = b;
}
else if(c > max) {
    max = c;
}
else{
    max = d;
}
System.out.println("max is " + max);

}
}

```

Enter a: 56

Enter b: 45

Enter c: 34

Enter d: 6

max is 6 (Wrong)

Max min mid ekai hoyanna

```

class Main {
    public static void main(String[] args) {
        int number1 = 10;
        int number2 = 70 ;
        int number3 = 54;

        int min=0,mid=0,max=0;

        if(number1 >= number2 && number1 >= number3){
            max = number1;
            if(number2 >= number3){
                mid = number2;
                min = number3;
            }
            else{
                mid = number3;
                min = number2;
            }
        }
    }
}

```

```

        else if(number2 >= number3 && number2 >= number1) {
            max = number2;
            if(number1 >= number3) {
                mid = number1;
                min = number3;
            }
            else{
                mid = number3;
                min = number1;
            }
        }
        else{
            max = number3;
            if(number2 >= number1) {
                mid = number2;
                min = number1;
            }
            else{
                mid = number1;
                min = number2;
            }
        }

        System.out.println(min + " , "+mid+" , "+max);
    }
}

```

Awarohana piliwela

```

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

        int n1 = 18;
        int n2 = 15;
        int n3 = 768758;
    }
}

```

```
int max = n1;
if (n2>max) {
    max = n2;
} else if (n3>max) {
    max = n3;
}

int min= n1;
if (n2<min) {
    min = n2;
} else if (n3<min) {
    min = n3;
}

int mid= n1;
if (n2<max && n2>min) {
    mid = n2;
} else if (n3<max && n3>min) {
    mid = n3;
}

System.out.println(max + "    " + mid + "    " +min);

}
}
```

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

        int n1 = 18;
        int n2 = 15;
        int n3 = 768758;
        if (n1>n2 && n1>n3) {
            if (n2 < n3) {
```

```
        System.out.println(n1 + " " + n2 + " " +
n3);
    } else{
        System.out.println(n1 + " " + n3 + " " +
n2);
    }
} else if (n2>n1 && n2>n3) {
    if (n1 < n3) {
        System.out.println(n2 + " " + n1 + " " +
n3);
    } else{
        System.out.println(n2 + " " + n3 + " " +
n1);
    }
}
else {
    if (n1 < n2) {
        System.out.println(n3 + " " + n1 + " " +
n2);
    } else{
        System.out.println(n3 + " " + n2 + " " +
n1);
    }
}
}
```

Positive number print

Oawoth 1 ena widihtath hadanna nathnam awul Inisa

```
class Main {  
    public static void main(String[] args) {  
  
        int num = 0;  
  
        if (0==0){ // 0 nam  
            System.out.println(1);  
        }  
        else if(num>0){ // positive nam  
            System.out.println(2);  
        }  
        else { // negative nam  
            System.out.println(3);  
        }  
    }  
}
```

```

        System.out.println(num);
    }else{// negative num

        System.out.println(num*-1);
    }

}
}

```

♣ calculator ekak if else-if walin hadamu

```

import java.util.Scanner;

class Calculator {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter first number: ");
        double num1 = input.nextDouble();

        System.out.print("Enter second number: ");
        double num2 = input.nextDouble();

        System.out.print("Enter operator (+, -, *, /): ");
        char operator = input.next().charAt(0);

        double result;

        if (operator == '+') {
            result = num1 + num2;
            System.out.println("Result = " + result);

        } else if (operator == '-') {
            result = num1 - num2;
            System.out.println("Result = " + result);

        } else if (operator == '*') {
            result = num1 * num2;
            System.out.println("Result = " + result);

        } else if (operator == '/') {
            result = num1 / num2;
            System.out.println("Result = " + result);

        } else {
            System.out.println("Invalid operator");
        }
    }
}

```

```

        result = num1 * num2;
        System.out.println("Result = " + result);

    } else if (operator == '/') {
        if (num2 != 0) {
            result = num1 / num2;
            System.out.println("Result = " + result);

        } else {
            System.out.println("Error: Cannot divide by zero.");
        }
    } else {
        System.out.println("Invalid operator.");
    }
}
}

```

◆ SWITCH statement

switch-case Statement

```

switch (variable) {
    case value1:
        // code
        break;
    case value2:
        // code
        break;
    default:
        // if no case matches
}

```

```

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

        int x = 100;
        switch(x) {

            case 100:

```

```
        System.out.println("I'm 100");
    case 200:
        System.out.println("I'm 200");
    case 300:
        System.out.println("I'm 300");
    case 400:
        System.out.println("I'm 400");
    case 500:
        System.out.println("I'm 500");
    default:
        System.out.println("I'm default");
    }

}
}
I'm 100
I'm 200
I'm 300
I'm 400
I'm 500
I'm default
```

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

        int x = 1000;
        switch(x) {

    case 100:
        System.out.println("I'm 100");
    case 200:
        System.out.println("I'm 200");
    case 300:
        System.out.println("I'm 300");
    case 400:
        System.out.println("I'm 400");
    case 500:
        System.out.println("I'm 500");
    default:
        System.out.println("I'm default");
    }

}
}
I'm default
```

```

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

        int x = 200;
        switch(x) {

            case 100:
                System.out.println("I'm 100");
            case 200:
                System.out.println("I'm 200");
            case 300:
                System.out.println("I'm 300");
            case 400:
                System.out.println("I'm 400");
            case 500:
                System.out.println("I'm 500");
            default:
                System.out.println("I'm default");
        }
    }
}

```

I'm 200
I'm 300
I'm 400
I'm 500
I'm default

SWITCH with break

```

class Main {
    public static void main(String[] args) {
        int x = 200;
        switch(x) {
            case 100:
                System.out.println("I'm 100");
                break;
            case 200:
                System.out.println("I'm 200");
                break;
            case 300:
                System.out.println("I'm 300");
                break;
            case 400:
                System.out.println("I'm 400");
                break;
            case 500:
                System.out.println("I'm 500");
                break;
        }
    }
}

```

```

        default:
            System.out.println("I'm default");
            break;
    }
}
I'm 300

```

break noda break eke wade karaganna widiha

```

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

        // 1 lne ekak witharak nam liyanna ona break noda liyana widiha

        int x = 300;

        switch(x) {

            case 100 -> System.out.println("I'm 100");

            case 200 -> System.out.println("I'm 200");

            case 300 -> System.out.println("I'm 300");

            case 400 -> System.out.println("I'm 400");

            case 500 -> System.out.println("I'm 500");

            // me awasthawedi default jump statement eka danna awashya
            // wenne na

            // java 14 idan thamai mehema liyanna chnce eka hamba une
        }
    }
}
I'm 300

```

♣ Switch calculator

```

import java.util.Scanner;

class Example {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

```

```

System.out.print("Enter first number: ");
int num1 = input.nextInt();

System.out.print("Enter an operator (+, -, *, /) :");
String operator = input.next();

System.out.print("Enter second number: ");
int num2 = input.nextInt();

switch (operator) {
    case "+":
        System.out.println(num1 + num2);
        break;

    case "-":
        System.out.println(num1 - num2);
        break;

    case "*":
        System.out.println( num1 * num2);
        break;

    case "/":
        if (num2 != 0) {
            System.out.println((double) num1 / num2);

        } else {
            System.out.println("Error");
        }
        break;
    default:
        System.out.println("Error");
}

}

```

♣ Switch label කියන්නේ මොකක්ද?

switch statement එක් අනුලෝධයෙන **case value** එක තමයි **switch label** කියන්නේ.

උදා:

```
int day = 3;
switch (day) {
    case 1:
        System.out.println("Monday");
        break;
    case 2:
        System.out.println("Tuesday");
        break;
    case 3:
        System.out.println("Wednesday");
        break;
    default:
        System.out.println("Another day");
}
```

මෙක්

- **case 1:, case 2:, case 3:** කියනව තමයි switch labels.
- **default:** කියන්නෙන් එක special switch label එකක්

♣ Java label (L1: වගේ) මොකක්ද?

Java වල **label** කියන්නේ statement එකකට *name* එකක් දෙන්න use කරන special syntax එකක්:

```
java
CopyEdit
L1:
statement;
```

උදා:

```
L1: for (int i = 0; i < 3; i++) {
```

```
        System.out.println("i = " + i);  
    }  


---


```

✳️ Switch / loop එකට label එකක්දාලා control flow manage කරන්න

මෙකේ විශේෂ වටින අවස්ථාව:

- nested loops / nested switch වලින් break හෝ continue කරන්න.
 - වෙන level එකක loop එකක්/ switch එකක් stop කරන්න.
-

● switch එකට label එකක්දාන්න

```
outerSwitch: // <-- මෙන්න label එක  
switch (value) {  
    case 1:  
        System.out.println("One");  
        break outerSwitch; // label එකට break  
    case 2:  
        System.out.println("Two");  
        break;  
    default:  
        System.out.println("Other");  
}
```

මෙහිදී:

- `outerSwitch` කියලා switch statement එකට label එකක්දීලා තියෙනවා.
 - `break outerSwitch;` කියලා label එක mention කරලා switch statement එකෙන් entirely exit වෙනවා.
-

● nested loops වල භාවිතය

```
outerLoop:  
for (int i = 0; i < 3; i++) {  
    for (int j = 0; j < 3; j++) {  
        if (i == j) {  
            break outerLoop; // exit outer loop entirely  
        }  
        System.out.println("i=" + i + ", j=" + j);  
    }  
}  
System.out.println("Exited loops");
```

මෙහිදි:

- `break outerLoop;` කියන්නේ outer loop එකට assign කල label එකට break වෙන එක.
- මෙය nested loop එකේ තියෙද්දී very useful.

▲ ඇත්තටම switch label practical use එක

- `switch` statement එකට label එකක් දාන්න පුළුවන්.
- හඳුනු ඇත්තෙනු නො යුතු න්‍යායෝගික ප්‍රාග්ධන අඩුයි.
- nested loops + switch එකේ mix එකකදී sometimes use වෙනවා.

Summary

- Java එකේ `L1` : , `outerLoop` : , `outerSwitch` : වගේ label එකක් ඕනම statement එකට assign කරන්න පුළුවන්.
- `break label;` → label assign කල statement එකක් exit වෙන්න.

- `continue label;` → label assign කළ loop එකේ next iteration එකට
යන්න

◆Java Logical Operators

Symbol	Description
<code>&</code>	Logical AND
<code> </code>	Logical OR
<code>&&</code>	Short-circuit AND
<code> </code>	Short-circuit OR
<code>!</code>	NOT

●Logical AND (&)

Okkoma true nam true ekak hari false unoth false

```
class Main {
    public static void main(String[] args) {
        System.out.println(true & true); // true
        //           1 x 1 = 1
```

```

        System.out.println(true & false); // false
        //           1   x   0   =   0

        System.out.println(false & true); // false
        //           0   x   1   =   0

        System.out.println(false & false); // false
        //           0   x   0   =   0
    }
}

```

true
false
false
false

Okkoma true nam true ekak hari false unoth false

```

class Main {
    public static void main(String[] args) {
        System.out.println(true & true & true); // true
        System.out.println(true & true & false); // false
        System.out.println(false & true & true); // false
        System.out.println(true & false & true); // false
        System.out.println(false & false & true); // false
        System.out.println(false & false & false); // false
        System.out.println(true & true & true & true); // true
        System.out.println(true & false & true & true); // false
    }
}

```

true
false
false
false
false
false
true
false

```

class Main {
    public static void main(String[] args) {
        System.out.println(true & true); // true
        System.out.println(true & false); // false
        System.out.println(false & true); // false
        System.out.println(true & true & true); // true
        System.out.println(true & false & true); // false
        System.out.println(true & false & true & true); // false
        System.out.println(true & true & true & true); // true
    }
}

```

```
        System.out.println(true & true & true & true & true & true &
true & true & false); // false
    }
}

true
false
false
true
false
false
true
false
```

●Logical OR (|)

Eka thanaka hari true thinawa nam answer true, serama false nam pamanay false

```
class Main {
    public static void main(String[] args) {
        System.out.println(true | true); // true
        System.out.println(true | false); // true
        System.out.println(false | true); // true
        System.out.println(false | false); // false
    }
}

true
true
true
false
```

Eka thanaka hari true thinawa nam answer true, serama false nam pamanay false

```
class Main {
    public static void main(String[] args) {
        System.out.println(true | true | true); // true
        System.out.println(true | true | false); // true
        System.out.println(false | true | false); // true
        System.out.println(false | true | true); // true
    }
}
```

```
        System.out.println(false | false | false | true); //true
        System.out.println(false | false | false | false | false); // false
    }
}
true
true
true
true
true
false
```

```
class Main {
    public static void main(String[] args) {
        System.out.println(true | true); // true
        System.out.println(true | false); // true
        System.out.println(false | true); // true
        System.out.println(false | false); // false
        System.out.println(true | true | true); // true
        System.out.println(true | false | true); // true
        System.out.println(true | false | true | true); // true
        System.out.println(true | true | true | true); // true
        System.out.println(false | true | false | false | false | false | false); // true
    }
}
```

```
true
true
true
false
true
true
true
true
true
```

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

        int prf_marks = 95; // >= 75
        int dbms_marks = 90; // >= 50
    }
}
```

```

        System.out.println(prf_marks >= 75);
        System.out.println(dbms_marks >= 50);
        System.out.println((prf_marks >= 75) & (dbms_marks >= 50)); //deken
ekak hari fail unoth false dekama pass nam true

        if(prf_marks >= 75 & dbms_marks >= 50) {      //deken ekak hari fail
unoth false dekama pass nam true
            System.out.println("Passed");
        } else {
            System.out.println("Failed");
        }

    }
}
true
true
true
Passed

```

Uda gaana thawa poddak clear karaganna warrahan dala ahadili karagattha

```

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

        int prf_marks = 95; // >= 75
        int dbms_marks = 90; // >= 50

        System.out.println(prf_marks >= 75);
        System.out.println(dbms_marks >= 50);
        System.out.println((prf_marks >= 75) & (dbms_marks >= 50)); //deken
ekak hari fail unoth false dekama pass nam true

        if((prf_marks >= 75) & (dbms_marks >= 50)) {      //deken ekak hari
fail unoth false dekama pass nam true
            System.out.println("Passed");
        } else {
            System.out.println("Failed");
        }

    }
}
true
true
true

```

Passed

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

        int prf_marks = 95; // >= 75
        int dbms_marks = 40; // >= 50

        System.out.println(prf_marks >= 75);
        System.out.println(dbms_marks >= 50);
        System.out.println((prf_marks >= 75) & (dbms_marks >= 50)); // true
& false = false

        if((prf_marks >= 75) & (dbms_marks >= 50)) { // true & false =
false
            System.out.println("Passed");
        } else {
            System.out.println("Failed");
        }

    }
}
true
false
false
Failed
```

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

        int prf_marks = 95; // >= 75
        int dbms_marks = 40; // >= 50

        System.out.println(prf_marks >= 75);
        System.out.println(dbms_marks >= 50);
        System.out.println((prf_marks >= 75) & (dbms_marks >= 50)); // true
& false = true
```

```

        if((prf_marks >= 75) | (dbms_marks >= 50)) { // true | false =
true (passed)
            System.out.println("Passed");
        } else {
            System.out.println("Failed");
        }

    }
}

```

Mehema e result ekama ena widihara gahannath puluwa

```

class Main {
    public static void main(String[] args) {
        int prfMarks = 95; // >=75
        int dbmsMarks = 60; // >=50
        boolean isPrfPassed = prfMarks>=75;
        boolean isdbmsPassed = dbmsMarks>=50;

        if(isPrfPassed & isdbmsPassed) { // true & false -> false
            System.out.println("Passed!");
        } else {
            System.out.println("Failed!");
        }
    }
}

```

Passed!

Deken ekak hari pass nam pass karawanna ona widihata hadanna

```

class Main {
    public static void main(String[] args) {
        int prfMarks = 95; // >=75
        int dbmsMarks = 40; // >=50
        boolean isPrfPassed = prfMarks>=75;
        boolean isdbmsPassed = dbmsMarks>=50;

        if(isPrfPassed | isdbmsPassed) { // true | false -> true
            System.out.println("Passed!");
        } else {
            System.out.println("Failed!");
        }
    }
}

```

```
import java.util.Scanner;
class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.println("Enter subject 01: ");
        int subject_1 = input.nextInt();

        System.out.println("Enter subject 02: ");
        int subject_2 = input.nextInt();

        System.out.println("Enter subject 03: ");
        int subject_3 = input.nextInt();

        if((subject_1>35) & (subject_2>35) & (subject_3>35)) {
            System.out.println("Passed");
        } else {
            System.out.println("Failed");
        }
    }
}
```

Enter subject 01:

2

Enter subject 02:

24

Enter subject 03:

4

Failed

Ekaparak hari fail nam fail ena widihata hadanna

```
import java.util.Scanner;

class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
```

```

System.out.println("Enter subject 01: ");
int subject1 = input.nextInt(); // 36 -> F

System.out.println("Enter subject 02: ");
int subject2 = input.nextInt(); // 55

System.out.println("Enter subject 03: ");
int subject3 = input.nextInt(); // 35

if(subject1<35 | subject2<35 | subject3<35) { // false | false |
false -> false -> passed
                                // true | false |
false -> true -> Failed
    System.out.println("Failed!");
} else {
    System.out.println("Passed!");
}

}
}

```

Enter subject 01:

23

Enter subject 02:

76

Enter subject 03:

12

Failed!

```

import java.util.Scanner;
class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.println("Enter subject 01: ");
        int subject_1 = input.nextInt();
        System.out.println("Enter subject 02: ");
        int subject_2 = input.nextInt();
        System.out.println("Enter subject 03: ");
        int subject_3 = input.nextInt();
    }
}

```

```

        if((subject_1>100) | (subject_2>100) | (subject_3>100))
    {
        System.out.println("Invalid marks: Marks should be less than or
equal to 100.");
    } else {
        if((subject_1>35) & (subject_2>35) &
           (subject_3>35)) {
            System.out.println("Passed");
        } else {
            System.out.println("Failed");
        }
    }
}

```

Enter subject 01:

34

Enter subject 02:

24

Enter subject 03:

45

Failed

```

import java.util.Scanner;
class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.println("Enter subject 01: ");
        int subject_1 = input.nextInt();

        System.out.println("Enter subject 02: ");
        int subject_2 = input.nextInt();

        System.out.println("Enter subject 03: ");
        int subject_3 = input.nextInt();

        if((subject_1>100 | subject_1<0) | (subject_2>100 | subject_2<0) |
(subject_3>100 | subject_3<0)) {
            System.out.println("Invalid marks: Marks should be less than or
equal to 100.");
        } else {
            if((subject_1>35) & (subject_2>35) &
               (subject_3>35)) {
                System.out.println("Passed");
            } else {
                System.out.println("Failed");
            }
        }
    }
}

```

```
    }  
}
```

Enter subject 01:

43

Enter subject 02:

53

Enter subject 03:

25

Failed

•Short Circuit AND (&&)

```
class Main {  
    public static void main(String[] args) {  
  
        int a = 15;  
  
        System.out.println(a>20 && a>10); //false  
        //          0 ----- =0  
        System.out.println(a>10 && a>20 && a>23 && a>1); //false  
        //          1          0----- = 0  
    }  
}
```

false

false

```
class Main {  
    public static void main(String[] args) {  
        int a = 10;  
        int b = 20;  
  
        System.out.println(a>15 & b>15); // false  
        System.out.println(a>15 && b>15); // false  
  
        System.out.println(a>5 && b>25); // false
```

```
    }  
}
```

• Short Circuit OR (||)

```
class Main {  
    public static void main(String[] args) {  
        int a = 15;  
        System.out.println(a>10 || a>20); // true  
        //           1 ----- = 1  
        System.out.println(a>20 || a>10 ); // true  
        //           0       1 ----- = 1  
    }  
}
```

true

true

```
class Main {  
    public static void main(String[] args) {  
        int a = 50;  
        int b = 10;  
  
        System.out.println(a>15 | b>15); // true  
        System.out.println(a>15 || b>15); // true  
  
        System.out.println(a>500 || b>5); // true  
    }  
}
```

```
class Main {  
    public static void main(String[] args) {  
  
        int x=5,y=3;  
  
        if(x++ == 5 & y++ == 5) {  
        }  
  
        System.out.println(x + " " + y); // 6 4  
    }  
}
```

6 4

```
class Main {  
    public static void main(String[] args) {  
  
        int x=5, y=3;  
  
        if(x++ == 5 && y++ == 5) {  
        }  
        //&& di palamu expesision eka false wena nisa ethanin eha run wenne na  
        //false agayak nisa ema nisa mul agaya calculate karala anek agaya calculate  
        //karanne nathuwa ehenma daanawa  
        System.out.println(x + " " + y); // 6 4  
  
    }  
}
```

6 4

```
class Main {  
    public static void main(String[] args) {  
  
        int y=3, x=5;  
  
        if(y++ == 5 && x++ == 5) {  
        }  
  
        System.out.println(y + " " + x); // 4 5  
        //&& di palamu expesision eka false wena nisa ethanin eha run wenne na false  
        //agayak nisa ema nisa mul agaya calculate karala anek agaya calculate  
        //karanne nathuwa ehenma daanawa  
    }  
}
```

4 5

```
class Main {  
    public static void main(String[] args) {  
        int x=5, y=3;  
        if(y++ == 5 && x++ == 5) {  
        }  
        System.out.println(x + " " + y); // 5 4  
    }  
}
```

//5 4

```
class Main {
```

```

public static void main(String[] args) {
    int x=5,y=3;
    if(x++ == 5 || y++ == 5) {
    }
    System.out.println(x + " " + y); // 6 4
}
}

//6 4

```

```

class Main {
    public static void main(String[] args) {
        int x=5,y=3;
        if(x++ == 5 || y++ == 5) {
        }
/*mul expesion eka komath true nisa ilaga ewa komath true wena nisa ||
waladi mul value eke agaya witharak x++
widihata calculate wela anith agaya ignore karana nisa ehi agay (y=3)ema
widihata wadinawa = 6 3 */
        System.out.println(x + " " + y); // 6 4
    }
}

//6 3

```

•NOT Operator !

`!true -> false`

`!false -> true`

```

class Main {
    public static void main(String[] args) {
        System.out.println(true);
        System.out.println(!true);

        System.out.println(false);
        System.out.println(!false);
    }
}

true

```

false
false
true

```
class Main {  
    public static void main(String[] args) {  
        int x = 65;  
        if((x>75)) {  
            System.out.println("Sri lanka");  
        } else {  
            System.out.println("India");  
        }  
    }  
}
```

India

```
class Main {  
    public static void main(String[] args) {  
        int x = 65;  
        if(!(x>75)) {  
            System.out.println("Sri lanka");  
        } else {  
            System.out.println("India");  
        }  
    }  
}
```

Sri lanka

◆ Ternary operators

(condition) ? Expression if true : Expression if false

```
class Main {  
    public static void main(String[] args) {  
        int x = 5;  
        int y = 6;  
        int max = 0;  
        if (x > y) {  
            max = x;  
        } else {  
            max = y;  
        }  
        System.out.println(max);  
    }  
}
```

```
class Main {  
    public static void main(String[] args) {  
        int a = 30;  
        int b = 20;  
/* int max = 0;  
if(a>b) {  
max = a;  
} else {  
max = b;  
} */  
        int max = (a>b) ? a : b; //me (a>b) condition eka true nam a max  
wenawa, false nam b max wenawa  
        System.out.println(max);  
    }  
}
```

30

Two numbers Max value

```
class Main {
```

```

public static void main(String[] args) {
    int x = 5;
    int y = 6;
    /* int max = 0;
    if (x > y) {
        max = x;
    } else {
        max = y;
    }*/
    int max = (x>y) ? x : y ;
    // (true)? true : false;
    System.out.println(max);
}

```

6

```

class Main {
    public static void main(String[] args) {
        int x = 30;
        int y = 20;
        int max = 0;

        max = (x>y) ? x : y;
        System.out.println(max);
    }
}

```

```

class Main {
    public static void main(String[] args) {
        int a = 30;
        int b = 20;

        System.out.println((a>b) ? a : b);
    }
}

```

75 ta wadi nam pass nathnam fail denna

```

class Main {
    public static void main(String[] args) {
        int marks = 65;
        String grade = (marks>=75) ? "Passed" : "Failed";
        System.out.println(grade);
    }
}
//Failed

```

```

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

```

```

        int marks = 26; // >=75 -> A, >=65 -> B, >=55 -> C,>=35 -> S , F

        String grade = (marks>=75) ? "A" : (marks>=65) ? "B" : (marks>=55) ?
"C" : (marks>=35) ? "S" : "F";

        System.out.println(grade);
    }
}

```

F

```

import java.util.Scanner;
class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter Your Marks : ");
        int marks =input.nextInt();
        // >=75 -> A, >=65 -> B, >=55 -> C,>=35 -> S , F

        String grade =(marks>=0 && marks<=100) ?
                (marks>=75) ? "A"      //--|
                : (marks>=65) ? "B"      //  |
                : (marks>=55) ? "C"      //  |-Condition eka true
nam meka karanawa
                : (marks>=35) ? "S"      //  |
                : "F"                    //--|
                :"Invalid marks";       //Condition eka false
nam meka karanawa

        System.out.println("Your Grade is " + grade);
    }
}

```

Enter Your Marks : 57

Your Grade is C

```
import java.util.Scanner;
class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter Your Value 1 : ");
        int num1 = input.nextInt();

        System.out.print("Enter Your Value 2 : ");
        int num2 = input.nextInt();

        System.out.print("Enter Your Value 3 : ");
        int num3 = input.nextInt();

        System.out.print("Enter Your Value 4 : ");
        int num4 = input.nextInt();

        int max = num1;
        max = (num2 > max) ? num2 : max;
        max = (num3 > max) ? num3 : max;
        max = (num4 > max) ? num4 : max;

        System.out.println("Max Value is " + max);
    }
}
```

Enter Your Value 1 : 23

Enter Your Value 2 : 345

Enter Your Value 3 : 13

Enter Your Value 4 : 54

Max Value is 345

DAY -7

QUIZE

Objective

Write a simple Java program that acts as a basic calculator.

The calculator should,

1. Read **two numbers** from the user.
2. Read **one arithmetic operator** (+, -, *, or /) from the user.
3. Perform the operation based on the operator entered.
4. Display the result.

Sample input/output:

```
1 Enter the first number: 10
2 Enter an operator (+, -, *, /): +
3 Enter the second number: 5
4 Result: 10.0 + 5.0 = 15.0
```

Requirements,

- Use the **Scanner** class to read input from the user via the terminal.
- You must be able to read:
 - An **integer** (e.g. 100)
 - A **String** for the operator (e.g. +)
- You only need to perform **one operation at a time** (i.e., no chaining or repeating).
- You can use a simple **if-else** or **switch** statement to decide which operation to perform.

Support: How to read a single word from the Input,

```
1 import java.util.Scanner;
2
3 public class HelloWorld{
4     public static void main(String []args){
5         Scanner input = new Scanner(System.in);
6         System.out.print("Enter your name: ");
7         String name = input.nextLine(); // Read input as a String
8         System.out.println("Your name is " + name);
9     }
10 }
```

```
import java.util.Scanner;
```

```
class Example {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter first number: ");
        int num1 = input.nextInt();

        System.out.print("Enter an operator (+, -, *, /) :");
        String operator = input.next();

        System.out.print("Enter second number: ");
        int num2 = input.nextInt();

        switch (operator) {
            case "+":
                System.out.println(num1 + num2);
                break;

            case "-":
                System.out.println(num1 - num2);
                break;

            case "*":
                System.out.println( num1 * num2);
                break;

            case "/":
                if (num2 != 0) {
                    System.out.println((double) num1 / num2);

                } else {
                    System.out.println("Error");
                }
                break;
            default:
                System.out.println("Error");
        }

    }
}
```

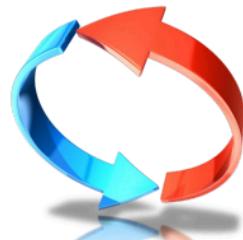
◆ Loops in Programming

Loops are fundamental **control structures** in programming that allow you to **repeatedly execute a block** of code as long as a **specified condition is met**.

Shortly ->

- control structure, repeatedly execute a block of code as long specified condition is met
- Loop is a control structure that repeats a block of code as long as a condition is true

Nishchitha condition ekak spuralana thuru block ekak repeat karana control structure ekak



```
class Main {  
    public static void main(String[] args) {  
  
        System.out.println(1);  
        System.out.println(2);  
        System.out.println(3);  
        System.out.println(4);  
        System.out.println(5);  
        System.out.println(6);  
        System.out.println(7);  
        System.out.println(8);  
        System.out.println(9);  
        System.out.println(10);  
    }  
}
```

```
}
```



```
1
```



```
2
```



```
3
```



```
4
```



```
5
```



```
6
```



```
7
```



```
8
```

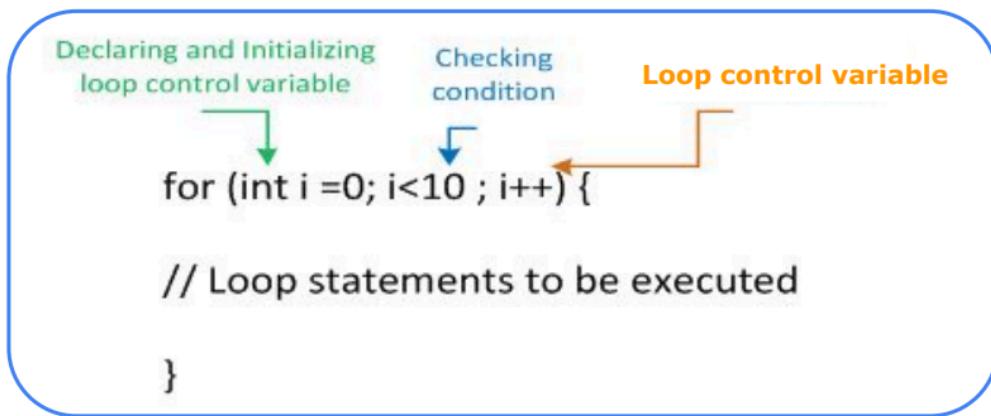


```
9
```



```
10
```

◆ For Loops



```
1          2          4  
for (initialization; termination; increment) {  
    3  
    // code to be executed  
}
```

1-> 2-> 3-> 4 -run wena piliwela

👉 එකම වරක් “condition” එකන්, “initialization” එකන්, “update” එකන් එකම line එකක ලියන loop එකක්.

👉 Loop එකේ Iteration ගණනක් (times) මූලම තීරණය වෙලා තියෙනවානම්, for loop වචා හොඳයි.

පදාහරණය:

```
for(int i = 1; i <= 5; i++) {  
    System.out.println(i);  
}
```

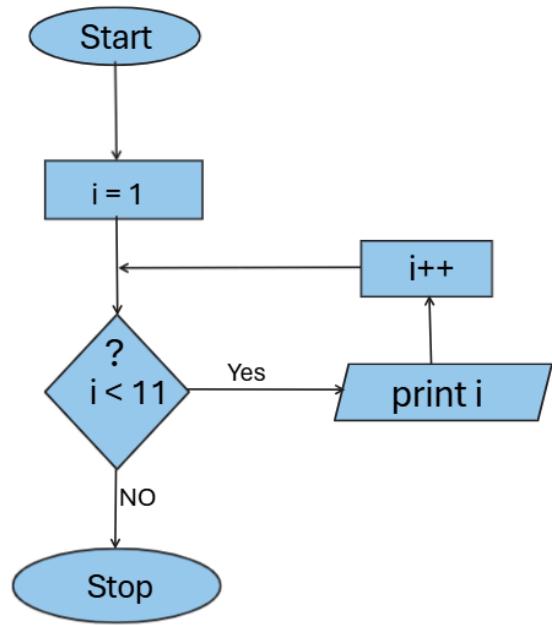
මේකේ:

- int i = 1; → initial value
- i <= 5; → condition
- i++ → update after each iteration

i<11 wenakalma me code eka repeat wenawa

```
class Main {  
    public static void main(String[] args) {  
        for(int i = 1; i<11; i++) {  
            System.out.println(i);  
        }  
    }  
}
```

1
2
3
4
5
6
7
8
9
10



Even values

```

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

        for(int i = 0; i<11; i+=2) {

            System.out.println(i);
        }
    }
}
  
```

0
2
4
6
8
10

Even values

```

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

        for(int i = 1; i<11; i++) {

            System.out.println(i);
        }
    }
}
  
```

2

4
6
8
10

Even numbers

```
class Main {  
    public static void main(String[] args) {  
  
        for(int i = 1; i<11; i++) {  
            if (i%2==0){  
                System.out.println(i);  
            }  
  
        }  
    }  
}
```

Odd numbers

```
class Main {  
    public static void main(String[] args) {  
  
        for(int i = 1; i<11; i++) {  
            if (i%2==0){  
  
            }else {  
                System.out.println(i);  
            }  
  
        }  
    }  
}
```

1
3
5
7
9

Odd number

```
class Main {  
    public static void main(String[] args) {  
  
        for(int i = 1; i<11; i++) {  
            if (i%2!=0){  
                System.out.println(i);  
            }  
        }  
    }  
}
```

```
        }
    }
}

1
3
5
7
9
```

Odd number

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

        for(int i = 1; i<11; i++) {
            System.out.println(i++);
        }
    }
}
```

5 gunakara

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

        for(int i = 1; i<21; i++) {
            if(i%5==0){
                System.out.println(i);
            }
        }
    }
}

5
10
15
20
```

5 gunakara

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

        for(int i = 1; i<21; i++) {
            i=i+4;
            System.out.println(i);
        }
    }
}
```

```
        }
    }
}
5
10
15
20
```

5 gunakara

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

        for(int i = 5; i<21; i+=5) {

            System.out.println(i);
        }
    }
}
```

```
5
10
15
20
```

1-10 athara 10sita 1 dakwa

```
class Main {
    public static void main(String[] args) {
        for(int i = 10; i>0; i--) {
            System.out.println(i);
        }
    }
}
```

```
10
9
8
7
6
5
4
3
2
1
```

```
class Main {
    public static void main(String[] args) {
        for(int i = 0; i<5; i++) {
```

```
        System.out.println("hello!");  
    }  
}  
}
```

hello!

hello!

hello!

hello!

Hello!

```
class Main {  
    public static void main(String[] args) {  
  
        for(int i = 1; i<11; i++) {  
            System.out.println("i love you");  
        }  
    }  
}
```

i love you

For eka athule thiyena int eliyata ganna puluwan me widihata

```
class Main {  
    public static void main(String[] args) {  
        int i = 0;  
        for(; i<10; i++) {  
            System.out.println("hello java!");  
        }  
    }  
}
```

hello java!

hello java!

hello java!

hello java!

hello java!

hello java!

```
hello java!  
hello java!  
hello java!  
hello java!
```

For eka athule thienā i=0 ha i++ mehema eliyata ganna puluwan

```
class Main {  
    public static void main(String[] args) {  
        int i = 0;  
        for(; i<10;) {  
            System.out.println("hello java!");  
            i++;  
        }  
    }  
}
```

```
hello java!  
hello java!
```

♣ infinite loop

computer program that repeats indefinitely because its termination condition is never met

එහි අවසන් කිරීමේ කොන්දේසිය කිසි විටෙකත් සපුරා නොමැති තිසා දින නියමයක් නොමැතිව පූනරාවර්තනය වන පරිගණක වැඩිසටහනක්

Never Ending Loop, never ending part eka nathi nisa

```
class Main {  
    public static void main(String[] args) {  
        for(int i=0; ; i++) {  
            System.out.println(i);  
        }  
    }  
}
```

2

..

..

No limit

Never Ending Loop

```
class Main {  
    public static void main(String[] args) {  
        int i = 0;  
        for(; i<10;) {  
            System.out.println(i);  
        }  
    }  
}
```

0

0

..

..

No limit

Never Ending Loop

```
class Main {  
    public static void main(String[] args) {  
        int i = 0;  
        for(; ;) {  
            System.out.println(i);  
        }  
    }  
}
```

0

0

..

..

No limit

```
class Main {  
    public static void main(String[] args) {  
        for(int i = 0; i<5; i++) {
```

```
        System.out.println(i);
    }
}
0
1
2
3
4
```

```
class Main {
    public static void main(String[] args) {
        for(int i = 0; i<5; i++) {
            System.out.println(i + 1);
        }
    }
}
1
2
3
4
5
```

```
class Main {
    public static void main(String[] args) {
        for (int i = 1; i < 6; i++) {
            System.out.println(i);
        }
    }
}
1
2
3
4
5
```

```
class Main {  
    public static void main(String[] args) {  
        for(int i = 0; i<10; i=i+2) {  
            System.out.println(i);  
        }  
    }  
}
```

0
2
4
6
8

```
class Main {  
    public static void main(String[] args) {  
        int sum = 0;  
        for(int i = 1; i<=10; i++) {  
            sum = sum + i;  
        }  
        System.out.println(sum);  
    }  
}
```

55

```
class Main {  
    public static void main(String[] args) {  
        int sum = 0;  
        for(int i = 1; i<=100; i++) {  
            sum = sum + i;  
        }  
        System.out.println(sum);  
    }  
}
```

5050

Meke apata ona answer eka enne na, ai e (sum eka nemei enne)

Mehema wenne for loop eka athule api variable declare karanne nathi nisa, api declare karanawa nam for lpop ekata kalin variable ekak widihata declare karala

eekata empty value ekak initialize karala ema variable name eka use karanawa for loop eka athule

```
class Main {  
    public static void main(String[] args) {  
  
        for(int i = 1; i<=10; i++) {  
            int sum = 1;  
            sum = sum + i;  
            System.out.println(sum);  
        }  
    }  
}  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11
```

Max Value

```
class Main {  
    public static void main(String[] args) {  
        int max = 0; // initial value  
  
        for (int i = 1; i <= 10; i++) {  
            if (i > max) {  
                max = i;  
            }  
        }  
  
        System.out.println("Max value inside the loop is: " +  
max);  
}
```

```
    }  
}
```

Q. Write a Java program to input 10 marks (type int) for subjects from the keyboard, find the total and average of the marks. Your output should be as follows:

Total : 756

Max : 98

Min : 67

Average : 75.6

1 kramaya

```
import java.util.Scanner;  
class Main{  
    public static void main(String args[]){  
        Scanner input = new Scanner(System.in);  
  
        int total = 0;  
  
        System.out.print("Enter Your Marks 1: ");  
        int marks = input.nextInt();  
  
        int max = marks;  
        int min = marks;  
  
        for (int i = 2 ; i<=10 ; i++){  
            System.out.print("Enter Your Marks "+ i +": ");  
            marks = input.nextInt();  
            if (marks > max) max = marks;  
            if (marks < min) min = marks;  
        }  
        System.out.println("Total Marks: " + total);  
        System.out.println("Average Marks: " + (total / 10));  
        System.out.println("Maximum Marks: " + max);  
        System.out.println("Minimum Marks: " + min);  
    }  
}
```

```

        if (marks>max) {
            max = marks;
        }

        if (marks<min) {
            min = marks;
        }

        total = total + marks;
    }
    System.out.println("Total :" + total);

    System.out.println("Max :" + max);
    System.out.println("Min :" + min);

    System.out.println("Avarage :" + (total/10.0));

}
}

```

Enter Your Marks 1: 12
Enter Your Marks 2: 34
Enter Your Marks 3: 23
Enter Your Marks 4: 12
Enter Your Marks 5: 23
Enter Your Marks 6: 34
Enter Your Marks 7: 34
Enter Your Marks 8: 3
Enter Your Marks 9: 98
Enter Your Marks 10: 87
Total: 360
Max: 98
Min: 3
Average: 36.0

2 kramaya

```

import java.util.Scanner;
class Main{
    public static void main(String args[]){
        Scanner input = new Scanner(System.in);

        int total = 0;

        int max = Integer.MIN_VALUE;
        int min = Integer.MAX_VALUE;

        for (int i = 1 ; i<=10 ; i++){
            System.out.print("Enter Your Marks "+ i +": ");
        }
    }
}

```

```

        int marks = input.nextInt();

        if (marks>max) {
            max = marks;
        }

        if (marks<min) {
            min = marks;
        }

        total = total + marks;
    }
    double average = total / 10.0;

    System.out.println("Total: " + total);
    System.out.println("Max: " + max);
    System.out.println("Min: " + min);
    System.out.println("Average: " + average);

}
}

```

Enter Your Marks 1: 12
Enter Your Marks 2: 34
Enter Your Marks 3: 23
Enter Your Marks 4: 12
Enter Your Marks 5: 23
Enter Your Marks 6: 34
Enter Your Marks 7: 34
Enter Your Marks 8: 3
Enter Your Marks 9: 98
Enter Your Marks 10: 87
Total: 360
Max: 98
Min: 3
Average: 36.0

♣ Factorial

```

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

        long fact = 1;
        for (int i = 5; i > 0; i--) {
            fact *= i;
        }
    }
}

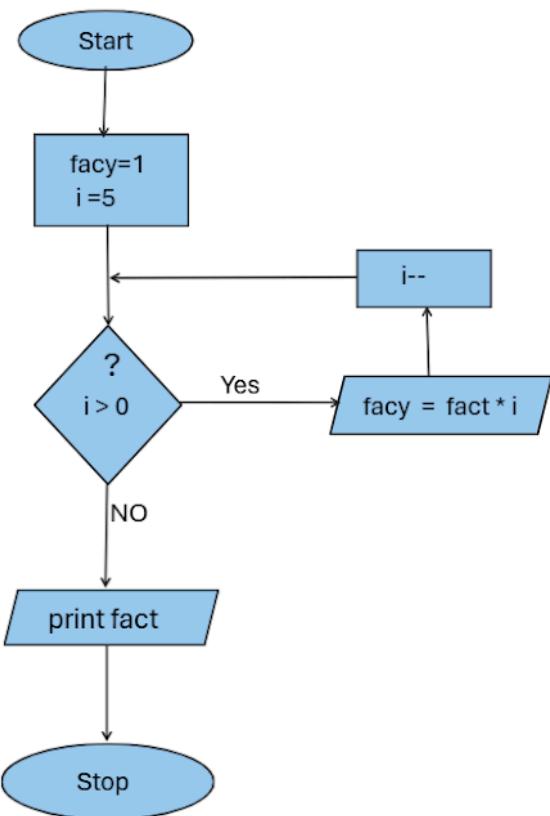
```

```

        System.out.println("Factorial of 5 is: " + fact);

    }
}

```



```

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

        int fact = 1;

        for(int i = 1 ; i<=5 ; i++) {
            fact*=i; // fact = fact*i; //6=6x5x4x3x2x1
            //    1f   = 1    x1
            //    2f   = 1    x2
            //    3f   = 2    x3
            //    4f   = 6    x4
            //    5f   = 24   x5

        }
        System.out.println("Factorial = " + fact);
    }
}

```

```
    }
}
```

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

        long fact = 1;
        for(int i = 1; i <= 3; i++){
            fact *= i;
        }
        System.out.println("Factorial of 3 is " + fact);
    }
}
```

Factorial of 10 is 6

```
import java.util.Scanner;

class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter a number : ");
        int number = input.nextInt();

        int fact = 1;

        for (int i = number; i > 0; i--) {
            fact *= i;
```

```

        }

        System.out.println("Factorial of " + number + " is: " +
fact);

    }

}

```

```

import java.util.Scanner;

class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter a number : ");
        int number = input.nextInt();

        long fact = 1;

        for (int i = number; i > 0; i--) {
            fact *= i;
        }

        System.out.println("Factorial of " + number + " is: " +
fact);

    }

}

```

Api input part eka iwara unama code eka awasanaye `input.close();` kiyala daanawa input part eka close karanna, mokada api ehema karanne scanner ekata ram eke yam idak ganna nisa

```

import java.util.Scanner;

class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

```

```

System.out.print("Enter a number : ");
int number = input.nextInt();

int fact = 1;

for (int i = number; i > 0; i--) {
    fact *= i;
}

System.out.println("Factorial of " + number + " is: " +
fact);

input.close();
}
}

```

Mulinma enter karana number eke factorial numberek balala ita passe 1-10 factorial wena wenama ganna

```

import java.util.Scanner;
class Main{
    public static void main(String args[]){
        Scanner input = new Scanner(System.in);

        System.out.print("Enter number : " );
        int num = input.nextInt();

        int fact = 1;

        for (int i = 1; i<=num; i++) {

```

```

        fact = fact * i ;
    }

System.out.println("factorial of :" + fact);

System.out.println("\nFactorials of numbers from 0 to
10:");

for (int j = 1 ; j<=10 ; j++) {
    int factorial = 1;
    for (int k=1 ; k<=j ; k++ ) {
        factorial = factorial * k;
    }
    System.out.println("factorial "+j+" is
"+factorial);
}

// for (int j = 1 ; j<=10 ; j++) {
//     long factorial = 1;
//     for (int k=j ; k>0 ; k-- ) {
//         factorial = factorial * k;
//     }
//     System.out.println("factorial "+j+" is
"+factorial);
// }

}

}

```

```

import java.util.Scanner;
class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter a number 1: ");
        int num1 = input.nextInt();

        System.out.print("Enter a number 2: ");
        int num2 = input.nextInt();

        System.out.println("sum is :" + (num1 + num2));
    }
}

```

```
    }  
}
```

♣PRIME Number (prathamaka sankya)

1 ha ema agayenm pamanak bedena sankya = prime nmber

1th 100th athara prime numbers

```
class Main{  
    public static void main(String args[]) {  
  
        for (int num=2; num <= 100 ;num++) {  
  
            boolean prime = true;  
  
            for (int i=2;i<num ; i++) {  
                if (num % i == 0){  
                    prime = false;  
                    break; //ekak hari false wenawa nam wadak  
nathi nisa  
                }  
  
            }  
            if (prime) {  
                System.out.println(num);  
            }  
        }  
    }  
}  
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
```

```
public class Main {  
    public static void main(String[] args) {  
        System.out.println("Prime numbers between 1 and 100:");  
    }  
}
```

```

// 2 සිට 100 දක්වා සංඛයා අතර loop එකක්
for (int num = 2; num <= 100; num++) {
    int i;

    // 2 සිට num-1 දක්වා divisor (බෙදන්න පුළුවන් ගණනක්) තිබේද
    කියලා බලන්න
    for (i = 2; i < num; i++) {
        if (num % i == 0) {
            // num එක i වලින් බෙදීය හැකි නම්, prime number එකක් නොවේ
            break; // loop එක නවන්වන්න
        }
    }

    // i == num නම්, කිසිම divisor එකක් භාවිත වී තැහැ; එනිසා
    num එක prime
    if (i == num) {
        System.out.print(num + " ");
    }
}
}

```

👉 වැඩි කරන හැටි :

2 සිට 100 දක්වා එක් එක් num වෙනුවෙන් පරික්ෂා කරනවා.

num එකට 2 සිට (num-1) දක්වා divisor එකක් තියෙනවද බලනවා.

තියෙනවනම් num % i == 0 වෙනවා, එහෙම උනොත් break කරනවා.

divisor එකක් භාවුවෙන් තැනි නම් (i == num), ඒ නුත් එක prime කියලා print කරනවා.

● Number ekak input karama e pramane input karama sum eka ena widihata hadanna

```

import java.util.Scanner;

class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter number count: ");
    }
}

```

```
int count = input.nextInt();

int sum = 0;

for (int i = 1; i <= count; i++) {
    System.out.print("Enter number " + i + ": ");
    int num = input.nextInt();
    sum = sum + num;
}

System.out.println("Sum is: " + sum);

input.close();
}
```

Enter number count: 3

Enter number 1: 2

Enter number 2: 2

Enter number 3: 3

Sum is: 7

◆ While Loops

```
while (condition) {
    // Code to execute while condition is true
    // Update condition (to avoid infinite loop)
}
```

- 👉 මූලින් condition එක check කරලා, true නම් body එක execute කරනවා.
- 👉 Iteration ගණනක් පුරුවලෙලාවට තීරණය නොවී, condition එක true වෙන තුරු දිගටම repeat කරන්න ඕනෑම යොදා ගන්නවා.

උදාහරණය:

```
int i = 1;
while(i <= 5) {
    System.out.println(i);
    i++;
}
```

මේක්:

- condition එක `i <= 5` true වෙයි නම් body එක් code run වෙනවා.
- false උනොත් body එකට යන්නේ තැහැ.

```
class Main {
    public static void main(String[] args) {
/*
for(int i=1; i<=10; i++) {
System.out.println(i);
}
*/
        int i = 1;
        while(i<=10) {
            System.out.println(i);
            i++;
        }
    }
}
```

1
2
3
4
5
6
7

8
9
10

```
class Main {  
    public static void main(String[] args) {  
  
        int i = 10;  
        while(i>0) {  
            System.out.println(i);  
            i--;  
        }  
    }  
}
```

10
9
8
7
6
5
4
3
2
1

Hodama widiha

```
class Main {  
    public static void main(String[] args) {  
  
        int i = 5;  
        while(i<101) {  
            System.out.println(i);  
            i+=5;  
        }  
    }  
}
```

5
10
15
20
25
30
35
40
45
50

55
60
65
70
75
80
85
90
95
100

5 gunakara hodama widiha nemei

```
class Main {  
    public static void main(String[] args) {  
  
        int i = 0;  
  
        while(i<100) {  
            i+=5;  
  
            System.out.println(i);  
        }  
    }  
}
```

sum

```
class Main {  
    public static void main(String[] args) {  
        int i = 1;  
        int sum = 0;  
        while(i<1001) {  
            sum = sum + i;  
            i++;  
        }  
        System.out.println("Sum = " + sum);  
    }  
}
```

Sum = 500500

♣Factorial

```
class Main {  
    public static void main(String[] args) {  
        int i = 1;  
        int fact = 1;  
        while(i<6) {  
            fact*=i; // fact = fact*i;  
            i++;  
        }  
        System.out.println("Factorial = " + fact);  
    }  
}
```

Factorial = 120 (factorial 5 =120)

```
class Main {  
    public static void main(String[] args) {  
  
        int fact = 1;  
  
        for(int i = 1 ; i<6 ; i++) {  
            fact*=i; // fact = fact*i; //6=6x5x4x3x2x1  
            // 1f = 1 x1  
            // 2f = 1 x2  
            // 3f = 2 x3  
            // 4f = 6 x4  
            // 5f = 24 x5  
  
        }  
        System.out.println("Factorial = " + fact);  
    }  
}
```

```
class Main {  
    public static void main(String[] args) {  
        int i = 5;  
        int fact = 1;  
        while(i>0) {  
            fact*=i; // fact = fact*i;  
            i--;  
        }  
        System.out.println("Factorial = " + fact);  
    }  
}
```

Factorial = 120

```

import java.util.Scanner;
class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter a number: ");
        int num = input.nextInt();

        int i = 1;
        int fact = 1;

        while (i <= num) {
            fact *= i;
            i++;
        }

        System.out.println("Factorial = " + fact);
    }
}

```

```

import java.util.Scanner;
class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter a number: ");
        int num = input.nextInt();

        int fact = 1; //5 x 4 x 3 x 2 x 1
        int i = num; //5

        while (i > 0) {
            fact *= i;
            i--;
        }

        System.out.println("Factorial = " + fact);
    }
}

```

- Number ekak input karama e pramane input karama sum eka ena widihata hadanna

```

import java.util.Scanner;
class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter number count: ");
    }
}

```

```

int count = input.nextInt();
int i = 0;
int sum = 0;

while(i<count) {

    System.out.print("Enter number " + (i+1) + ":");
    int number = input.nextInt();

    sum = sum + number;

    i++;
}
System.out.println("Sum = " + sum);
}
}

```

Enter number count: 3

Enter number 1:2

Enter number 2:2

Enter number 3:2

Sum = 6

```

import java.util.Scanner;
class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter number count: ");

        int count = input.nextInt();
        int i = 0;
        int sum = 0;

        while(i<count) {
            System.out.print("Enter number " + (i+1) + ":");

            int number = input.nextInt();

            sum = sum + number;

            i++;
        }
        System.out.println("Sum = " + sum);
        input.close();
    }
}

```

Enter number count: 2

Enter number 1:2

Enter number 2:12

Sum = 14

Mul odd number 10 print karanna

```
class Main {  
    public static void main(String[] args) {  
  
        int num = 1;  
        int count = 1;  
  
        while(count <=10){  
            System.out.println(num);  
            num += 2;  
            count++;  
        }  
    }  
}
```

◆ Do-while Loops

```
do {  
    // Code to execute at least once  
    // Update loop variable (to avoid infinite loop)  
} while (condition);
```

- 👉 Body එක execute කරලා ඉදලා පස්සේ condition check කරනවා.
- 👉 ඒ නිසා අවම වගයෙන් එක වරක් body එක run වෙනවා.

උදාහරණය:

```
int i = 1; //do-while eke methana initialize karanna onama na  
do {  
    System.out.println(i);  
    i++;  
} while(i <= 5);
```

මෙක්:

- മുൻ്നിൽ System.out.println(i); execute ലേക്കാം.
- അജ്ഞേജ് i <= 5 check ലേക്കാം

```
class Main {
    public static void main(String[] args) {
        int i = 0;
        do {
            System.out.println(i); // 0
            i++;
        } while(i<10);
    }
}
```

0
1
2
3
4
5
6
7
8
9

```
class Main {
    public static void main(String[] args) {
        int i = 1;
        do {
            System.out.println(i); // 0
            i++;
        } while(i<=10);
    }
}
```

1
2
3
4
5
6

7
8
9
10

Even number ,Hoda widiha

```
class Main {  
    public static void main(String[] args) {  
        int i = 2;  
        do {  
            if (i%2==0) {  
                System.out.println(i);  
            }  
            i++;  
        }while (i<=10);  
    }  
}
```

2
4
6
8
10

```
class Main {  
    public static void main(String[] args) {  
        int i = 1;  
        do {  
            i++;  
            System.out.println(i);  
  
            i++;  
        }while (i<=10);  
    }  
}
```

2
4

6
8
10



සිරාංගය:

	for loop	while loop	do-while loop
Condition check කරන වෙළාව	මුලම	මුලම	body run කරාට පස්සේ
අවම iteration ගණන	0	0	1
හොඳම යොදා ගන්න තැන	Iteration ගණන fix නම්	Iteration ගණන නොදැනී	අවම වරක් run කරන්න ඕනම නම්



දිදාහරණය: “කමලා තේ බිලා ගෙදර යනවා”

කමලා ගෙදර යනකළේ තියා තේ බින්න හිතනවා කියලා හිතමු.

එශකට 3 loops වලින් කොහොම represent කරනවද බලමු:

● 1. for loop

දිදාහරණය:

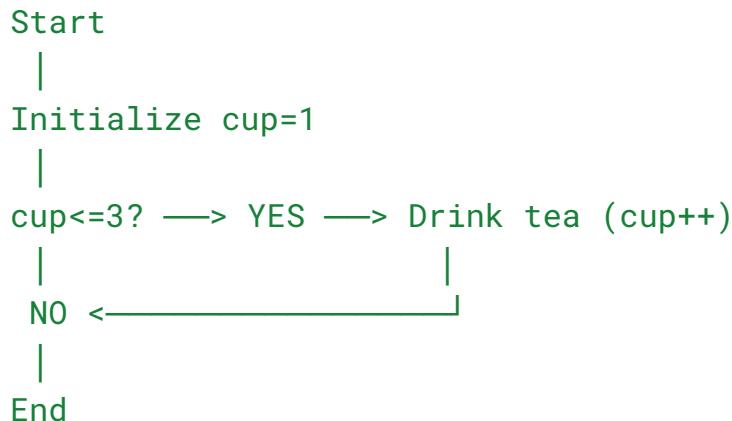
- කමලා කියනවා: “මම cups 3ක් පමණක් බින්නම්.”

```

for(int cup = 1; cup <= 3; cup++) {
    System.out.println("Cup " + cup + " of tea");
}

```

Diagram



★ එකම line එකකින්ම කී වරක් repeat කරනවද කියලා fix කරනවා.

```

class Example{

    public static void main(String[] args) {

        for(int i=0;i<10;i++) {}

        for(int i=0;i<10;i++) {}

    }
}

```

League

● 2. while loop

උදාහරණය:

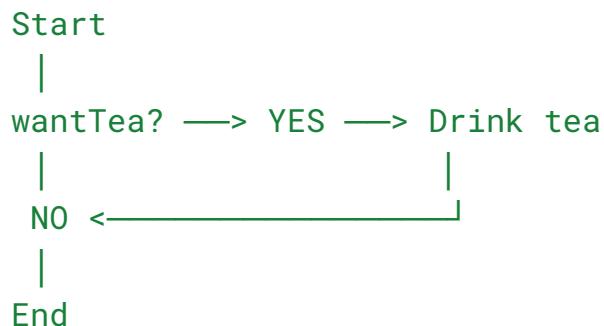
- කමලා කියනවා: “මම තව තේ ලින්න හිතෙන තුරු ලින්නම්.”

```

boolean wantTea = true;
while(wantTea) {
    System.out.println("Drink a cup of tea");
    // Imagine: after some cups, wantTea=false;
}

```

Diagram



★ මොන වතාවකද stop වෙනවද runtime එක් තීරණය වෙනවා.

● 3. do-while loop

උදාහරණය:

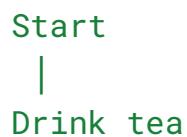
- කමලා කියනවා: “මම අවම වගයෙන් එක් cup එකක් නම් ලින්නම්. පස්සේ තව ලින්න හිතෙනවද බලන්නම්.”

```

boolean wantTea = false;
do {
    System.out.println("Drink a cup of tea");
} while(wantTea);

```

Diagram



```

| 
wantTea? —> YES —> Drink tea
|
NO
|
End

```

★ අවම වශයෙන් එක් cup එකක් ලිලා පස්සේ decide කරනවා තව බිත්තා හිතෙනවද කියලා.

- do-while loop එක් කොන්දේසිය පරීක්ෂා කරන එක loop එක් body එක ධාවනය කළ පසුවයි.
- ඒ නිසා body එක අනිවාර්යයෙන් එක වතාවක් ක්‍රියාත්මක වේ.

එමතිසා:

- පරීගීලකගෙන් data එකක් අනිවාර්යයෙන්ම මූලින්ම එක් වතාවක් ලැබිය යුතු විට,
- හෝ menu එකක් පළමුවන් පෙන්විය යුතු විට,

do-while loop එක වචා සූදුසුයි



සිංහලයි:

Loop	කී වරක් run වෙනවද fix?	අවම iteration	Use case
for	ඔව්	0	Iteration ගණන fixed
while	නැහැ	0	condition true වෙන තුරු
do-while	නැහැ	1	අවම වරක් run කරන්න ඕන

While | do while athara wenasa

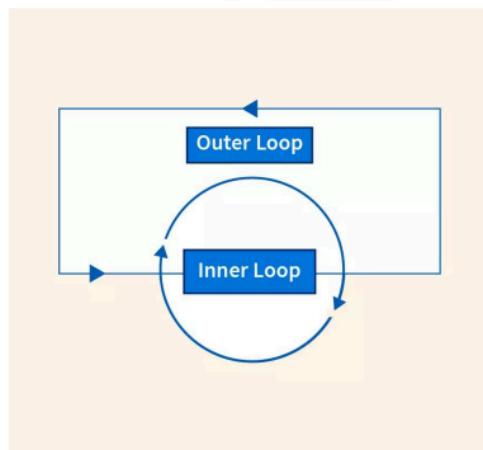
do-while loop එකෙන් condition එක පරීක්ෂා කරන එකට පෙර body එක බාවනය කරන නිසා, පලමු input එකක් ගැනීමට නොදැකී.

while loop එකෙන් condition එක මූලින්ම බලන නිසා, කෙලින්ම condition false නම් body එකකට යන්නේ නෑ.

◆ Nested Loops

placing one loop inside the body of another loop,

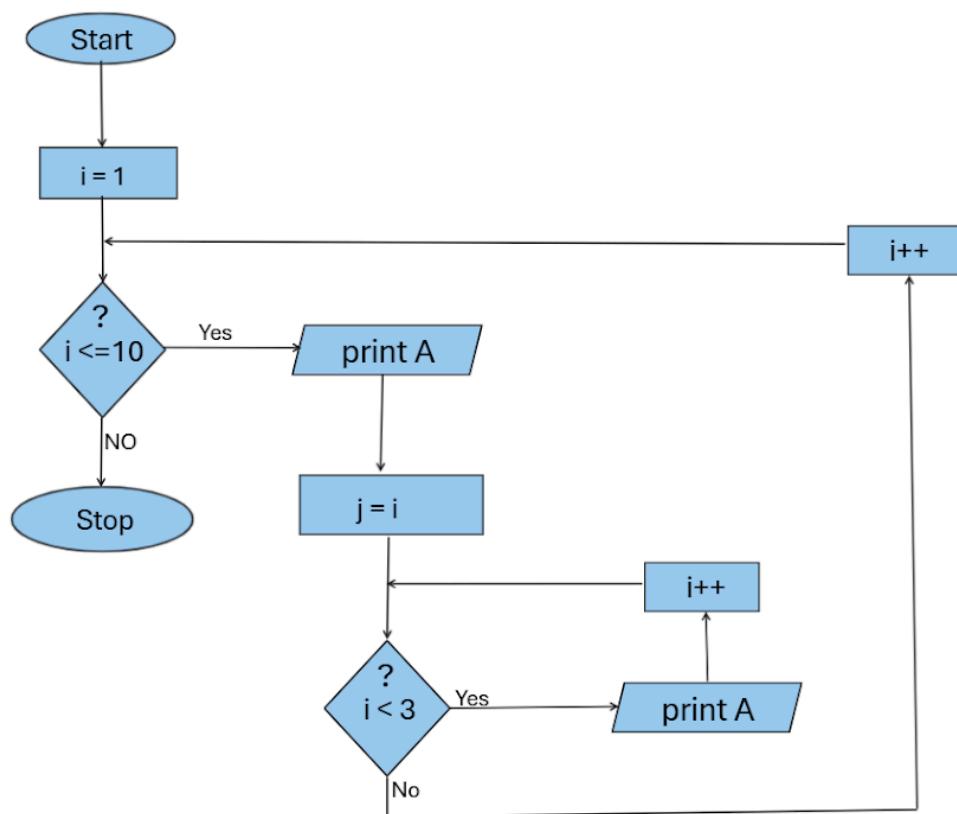
The inner loop runs completely every time the outer loop runs once
(බාහිර දුපය එක් වරක් බාවනය වන සිංහ අවස්ථාවකම අභ්‍යන්තර දුපය සම්පූර්ණයෙන්ම කියාත්මක වේ.)



```
class Main {  
    public static void main(String[] args) {  
        for(int i=0; i<5 ; i++) {  
            System.out.println("A");  
            for(int j=0; j<3; j++) {  
                System.out.println("B");  
            }  
        }  
    }  
}
```

```
    }  
}  
}
```

A
B
B
B
A
B
B
B
A
B
B
B
A
B
B
B



HW

```
class Main {  
    public static void main(String[] args) {  
        for(int i = 1; i<7; i++) {  
            for(int j = 0; j<i; j++) {  
                System.out.print("* ");  
            }  
            System.out.println("");  
        }  
    }  
}  
*  
* *  
* * *  
* * * *  
* * * * *
```

```
class Main {  
    public static void main(String[] args) {  
        for(int i = 6; i>0; i--) {  
            for(int j=0; j<i; j++) {  
                System.out.print("* ");  
            }  
            System.out.println("");  
        }  
    }  
}  
* * * * *  
* * * *  
* * *  
* *  
*
```

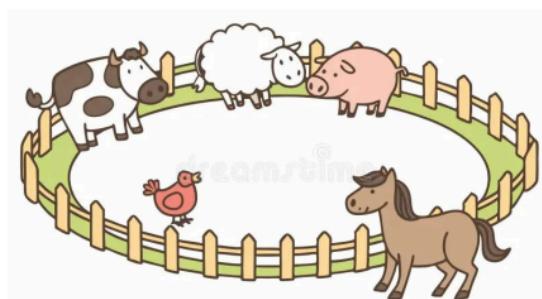
```
class Main {  
    public static void main(String[] args) {  
        for(int i = 1; i<=6; i++) {  
            for (int k = 6- i; k>=0 ; k--) {  
                System.out.print(" ");  
            }  
        }  
    }  
}
```

```
        for(int j = 0; j<i; j++) {
            System.out.print(" * ");
        }
        System.out.println("");
    }
}
*
*
*
*****
*****
*****
```

```
class Main {
    public static void main(String[] args) {
        for(int i = 6; i>0; i--) {
            for (int k = 0; k < 6-i; k++){
                System.out.print("  ");
            }
            for(int j=0; j<i; j++) {
                System.out.print(" * ");
            }
            System.out.println("");
        }
    }
}
*****
*****
*****
*****
*
```

◆ Variable Scope

As a general rule, variable that are defined within a block are not accessible outside the block.



Wurkayata wata athule inna saththu kanna baa wage, wata athule inna bataluwō Variable scope wage.

Pita inna wurkayata, wata athule inna bataluwonta balapamak ella karanna ba.

```
class Main {  
    public static void main(String[] args) {  
        int x = 100;  
        {  
            int y = 200;  
            System.out.println(y);  
        }  
        int z = 300;  
        System.out.println(x);  
        System.out.println(y); // Illegal  
        System.out.println(z);  
    }  
}
```

Variable Scope

Variable Scope

```
class Main {  
    public static void main(String[] args) {  
        int x = 100;  
        {  
            int y = 200;  
            System.out.println(y);  
        }  
        int z = 300;  
        System.out.println(x);  
        System.out.println(y); // Illegal  
        System.out.println(z);  
    }  
}
```

Red color block eken eiye y wa acces karanna ba eyawa imit wenne block eka athule witarai eyawa access wenne block eka athule idagena witharai

Habai, block eken eliye variable ekak declare karala eka block eka athule hadanna puluwan, apata block eka arthur ewa eliyata acces karanna ba

Eyawa acces karanna puluwan area eka scope eka

- Block eka athule ewata block eken eliye ewa access karanna puluwan
- Block eken pita hadapu ewat block eka athule ewa access karann ba

Java code ekak run wenne uda idan pahalata

Explain variable scope

```
class Main {  
    public static void main(String[] args) {  
        int x = 100;  
        System.out.println(x);  
        System.out.println(y); // illegal (y hadala na sop ekata kalin)  
        System.out.println(z); // illegal (z hadala na sop ekata kalin)  
  
        //variable scope start  
        {  
            int y = 200;  
            System.out.println(x); // block eken eliye ewa block eka athule  
use karanna puluwan  
            System.out.println(y);  
            System.out.println(z); // illegal (z hadala na sop ekata kalin)  
        }  
        // variable scope end  
  
        int z = 300;
```

```
    System.out.println(x);
    System.out.println(y); // illegal (y hadala na sop ekata kalin) y
hadala thiyyenne pita block ekaka nisa ema y adala wenne ema block ekata
witharai
    System.out.println(z);
}
}
```

Day 7

Quiz

1. Write a Java program using **nested for** loops to print the following pyramid star pattern,

```
*  
***  
*****  
*****  
*****
```

Instructions:

- You must use Java for loops to generate this output.
- The number of rows should be set using a variable (e.g., int rows = 5;).
- Do not use string formatting or external libraries, only loops and basic printing (System.out.print() and System.out.println()).

♣STARS CASE

```
class Main {  
    public static void main(String[] args) {  
  
        for (int i = 1; i <= 5; i++) {  
  
            // Print spaces  
            for (int j = 1; j <= 5 - i; j++)  
                System.out.print(" ");  
  
            // Print stars  
            for (int k = 1; k <= 2 * i - 1; k++) // (2n-1) = 1,  
3, 5, 7, 9...  
                System.out.print("*");  
  
            // New line  
            System.out.println();  
        }  
    }  
}  
*  
  
***  
*****  
*****  
*****
```

```
class Main{  
    public static void main(String args[]){  
        int rows = 5;  
  
        for (int i=1; i<=rows ; i++){  
            for (int j =i; j<=rows; j++ )  
                System.out.print(" ");  
  
            for (int k =1; k<=2*i-1; k++)  
                System.out.print("*");  
  
            System.out.println();  
        }  
    }  
}
```

```
*  
***  
*****  
*****  
*****
```

◆ Jump statements : “break” , “continue” , “return”

use for alter the normal flow of control

Program eke Condition flow eka galaayaama nawaththala flow eke wena thanak Run karanawa

```
class Example{  
    public static void main(String[] args) {  
        System.out.println("Start main");  
        for(int i = 0; i < 10; i++){  
  
            System.out.println("i : " + i);  
        }  
        System.out.println("End main");  
    }  
}
```

Start main

i : 0
i : 1
i : 2
i : 3
i : 4
i : 5
i : 6
i : 7
i : 8

```
i : 9  
End main
```

●break

Break eken wenne loop eke wena wada serama nawaththala ema loop/condition
eken eliyata enawa
condition true Loop eka nawaththanawa

break-> Exit the loop or the condition

5 == 5 unu gaman if statement eka athulata yanawa ,athule break thiinawa dakka
gaman loop eke wada nawaththala ema loop eken eliyata yanawa

```
class Example{  
    public static void main(String[] args) {  
        System.out.println("Start main");  
        for(int i = 0; i < 10; i++){  
            if(i == 5){  
                break;  
            }  
            System.out.println("i : " + i);  
        }  
        System.out.println("End main");  
    }  
}
```

```
Start main  
i : 0  
i : 1  
i : 2  
i : 3  
i : 4  
End main
```

```
class Main {  
    public static void main(String[] args) {  
        for(int i=1; i<11; i++) {  
            if(i%2==0) {  
                break;
```

```

        }
        int result = i * 100;
        System.out.println(result);
    }
    System.out.println("Hello");
}
}

```

100

Hello

●continue

Continue eken wenne adala condition eka true unu gaman continue nam thiennen Loop ekata adaala ema wataya ethanin nawaththala ilaga wataya patan gannawa

Loop/ condition eke condition true continue unu gamn wataya ethanin nawaththala ilaga wataye idan run karanawa

ඡාවා හි continue ප්‍රකාශය භාවිතා කරන්නේ ලුපයක වන්මත් ප්‍රකරාවර්තනය මග හැරීමටයි.

Condition -> skip the current iteration and move to next iteration of the loop

5 ==5 unu gaman if statement eka athulata yanawa athule continue thiinawa dakka gaman ema watayata adaala wada tika nokara (skip karala) ilaga wate idan nawatha wada karanna gannawa

```

class Example{
    public static void main(String[] args) {
        System.out.println("Start main");
        for(int i = 0; i < 10; i++){
            if(i == 5){
                continue;
            }
            System.out.println("i : " + i);
        }
        System.out.println("End main");
    }
}

```

Start main

i : 0

i : 1

i : 2

i : 3

i : 4

i : 6

```
i : 7  
i : 8  
i : 9  
End main
```

●Return

Return hambaunu gaman sampurna program eka nawathinawa, kiyala hithaganna(namuth eka practical naa)

Aththatama wenne method eken eliyata yana eka, methana main method eka witharak thina nisa methana e condition eka through eliyata yanawa

```
class Example{  
    public static void main(String[] args) {  
        System.out.println("Start main");  
        for(int i = 0; i < 10; i++){  
            if(i == 5){  
                return;  
  
            }  
            System.out.println("i : " + i);  
        }  
        System.out.println("End main");  
    }  
}
```

```
Start main  
i : 0  
i : 1  
i : 2  
i : 3  
i : 4
```

◆Java Methods

Java Methods are blocks of code that perform a specific task.

A method allows us to reuse code, improving both efficiency and organization.

All methods in Java must belong to a class.



- **Java method ekak kiyanne code block ekak eka liyala thiylene yam kisi specific task ekak perform karanna**

Liyala thiylene code block eka nawatha nawatha bawitha kirimata awasthawa thinawa

Karyakshamathawa saha organize karala thiylene widiha improve karanawa me methods kina aya

Methods apita hadanna puluwan class eka athule witharai
Ex: class Main{ java methods saha anekuth siyalla }

Java method ekak enne mewage,
Public void getvalue (int num1, int num2){
 //method body
}

Public - Modifier

- Methana nama wenas wenna puluwan
 - Modifiers kihipayak unath enna puluwan
 - Modifiers kotas dekak thiyenawa,
- Modifiers**
-



Static - Modifier

Void – Return type

- Methana nama wenas wenna puluwan

Getvalue - Method Name

- Methana nama wenas wenna puluwan

(int num1, int num2) - Parameter List

Int num1 - parameter

- Methanadi variable widihata wada karata api mewata kiyanne parameter kiyala
- Parameter nathi wennath puluwan, 1k thiyennath puluwan, kihipayak thiyennath puluwan

Getvalue (int num1, int num2) - method signature

Main method,

Public static void main(String argos[]){}

public - modifier

Static - modifier

Void - Return Type

Main - method name

(String args[]) - parameter list

◆ Types Of Java Methods

Predefined Methods

User-defined Methods



● 1. Predefined Methods (Built-in Methods)

- Java standers library samaga inbuild paminena methods
ex:- random

- මෙවා Java language එකේම ඉස්සෙල්ලම නිඩිල method.
- Java API (Application Programming Interface) එක තුළ ඇතුළත් වෙනවා.
- අපිට just call කරන්න පූලුවත්, redefine කරන්න නැ.

දදාහරණ:

```
System.out.println("Hello");
int len = "Hello".length();
double sq = Math.sqrt(25);
```

මෙහි **println()**, **length()**, **sqrt()**,**random()** — මෙවා **predefined methods**.

● 2. User-defined Methods

- Programmersla athin hadana methods

- Programmer එකා තමන්ගේ අවශ්‍යතාවයට අනුව නිර්මාණය කරන method එකක්.
- මෙය **අපිම define** කරලා, අවශ්‍ය වටා call කරනවා.

උදාහරණ:

```
public static void greet(String name) {  
    System.out.println("Hello " + name);  
}  
  
public static int add(int a, int b) {  
    return a + b;  
}
```

● Method name eka liyana standard eka

1.method name eka aniwaren patanganna ona simple letter ekakin

2.camel case widihata liyanna ona

(deweni word eke palaweni akura Capital wennaona)

Ex:- getValue

3.method name eka meaning full wenna ona

(karana specific task ekata adaalawa namak denna)

Ex:- ekathuwak nam sumValue kiyala denawa

4.method name start karanna ona verb ekakin

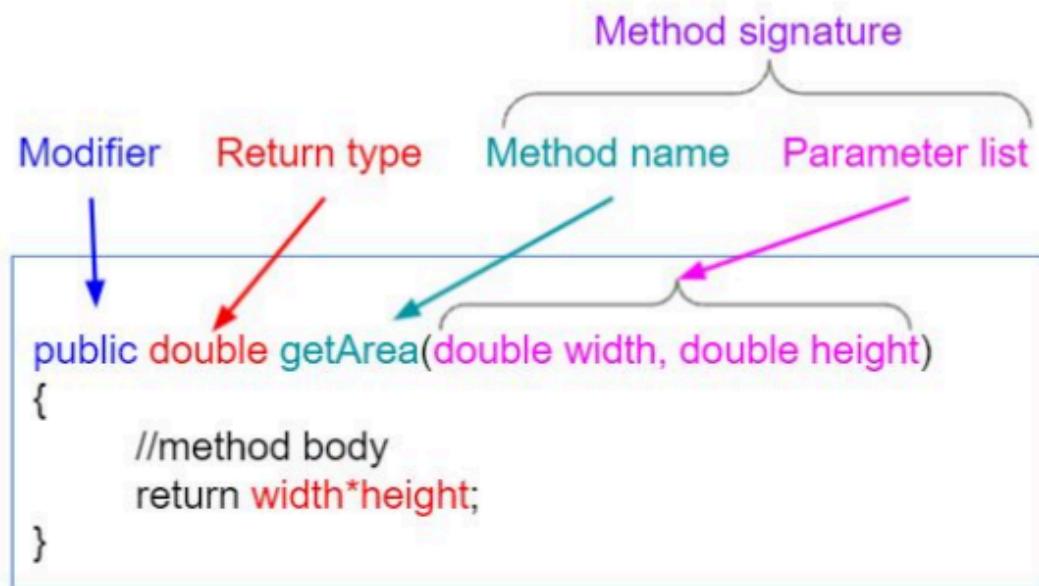
Ex:- `findMax`

Ehema verb ekakin patanganne nathi awasthath thinawa

Ex:- `main`

Namuth puluwan tharam try karanna standers ekak widihata verb ekakin start karala liyanna method name ekak.

●Anatomy of Java Methods



Method calling

```
class Main {
    // start main method
    public static void main(String[] args) {
        sing(); // method calling
        System.out.println("Do your exercises");
        System.out.println("Run");
        sing();
    }
    // End main method

    // start sub method
    public static void sing() {
        System.out.println("Warm up!");
        System.out.println("Music On");
        System.out.println("Sing your song");
    }
    // End sub method
}
```

Warm up!

Music On

Sing your song

Do your exercises

Run

Warm up!

Music On

Sing your song

Address eka parak print karaganna ona

```
class Main {  
    public static void main(String args[]) {  
        System.out.println("Kavindu Samarasinghe");  
        System.out.println("10 Station Rd");  
        System.out.println("Colombo 04");  
  
        System.out.println("Kavindu Samarasinghe");  
        System.out.println("10 Station Rd");  
        System.out.println("Colombo 04");  
  
        System.out.println("Kavindu Samarasinghe");  
        System.out.println("10 Station Rd");  
        System.out.println("Colombo 04");  
    }  
}
```

Kavindu Samarasinghe
10 Station Rd
Colombo 04
Kavindu Samarasinghe
10 Station Rd
Colombo 04
Kavindu Samarasinghe
10 Station Rd
Colombo 04

Address eka 3parak print karana eka java method valin

```
class Main {  
    public static void printMyAddress() {  
        System.out.println("Kavindu Samarasinghe");  
        System.out.println("10 Station Rd");  
        System.out.println("Colombo 04");  
    }  
    public static void main(String[] args) {  
        printMyAddress(); // method calling  
        printMyAddress(); // method calling  
        printMyAddress(); // method calling  
    }  
}
```

Kavindu Samarasinghe
10 Station Rd

Colombo 04
Kavindu Samarasinghe
10 Station Rd
Colombo 04
Kavindu Samarasinghe
10 Station Rd
Colombo 04

Onama java program ekaka mulinma run wenne mainmethod eka, main method ekak thiyenna puluhan java program ekakata ekai

```
class Main {  
    public static void printMyAddress() {  
        System.out.println("Kavindu Samarasinghe");  
        System.out.println("10 Station Rd");  
        System.out.println("Colombo 04");  
    }  
  
    public static void main(String[] args) {  
        System.out.println("=====");  
        printMyAddress();  
        System.out.println("=====");  
        printMyAddress();  
        System.out.println("=====");  
        printMyAddress();  
    }  
}
```

=====

Kavindu Samarasinghe

10 Station Rd
Colombo 04

=====

Kavindu Samarasinghe

10 Station Rd
Colombo 04

=====

Kavindu Samarasinghe
10 Station Rd
Colombo 04

Methana wenne,

Program eka run karama mulinma execute wenne main method eka, java uda idan pahalata thamai hamathissema execute wenne, apata mulinma enne hello kila print karanna, ita passe terminal eke hello kila print karanawa, ita pase print hi kina method eka call karanawa , ita passe e method eka athule thina code eka uda idan pahalata execute karala ethana hi kila print karanawa

```
class Main {  
    public static void printHi() {  
        System.out.println("Hi");  
    }  
  
    public static void main(String[] args) {  
        System.out.println("Hello"); // specific task eka  
        printHi(); //method calling  
  
        System.out.println("Welcome"); // specific task eka  
        printHi(); //method calling  
    }  
}
```

Hello
Hi
Welcome
Hi

Robot dance or sing

```
class Main {  
    public static void sing(){  
        System.out.println("Warm up");  
        System.out.println("Sing a Song");  
    }  
  
    public static void dance(){  
        System.out.println("Music On");  
        System.out.println("Dance");  
    }  
  
    public static void main(String[] args) {  
        sing();  
        dance();  
        dance();  
        sing();  
  
    }  
}
```

Warm up
Sing a Song
Music On
Dance
Music On
Dance
Warm up
Sing a Song

Tea Machine program

Java Methods



```
1- class Main {  
2-     public static void makeCupOfCoffee() {  
3-         System.out.println("Place a coffee filter in the basket");  
4-         System.out.println("Add 1-2 tbsp ground coffee per 6 oz water.");  
5-         System.out.println("Fill reservoir with water.");  
6-         System.out.println("Turn on coffee maker.");  
7-         System.out.println("Wait to brew.");  
8-         System.out.println("Pour and enjoy!");  
9-     }  
10-    public static void prepareCoconutSambol() {  
11-        System.out.println("Grate 1 cup coconut.");  
12-        System.out.println("Add onion, chili, salt.");  
13-        System.out.println("Mix in chili powder.");  
14-        System.out.println("Add lime juice.");  
15-        System.out.println("(Optional) Add Maldivian fish.");  
16-        System.out.println("Mix well.");  
17-    }  
18-    public static void main(String[] args) {  
19-        makeCupOfCoffee();  
20-        prepareCoconutSambol();  
21-    }  
22- }  
23-  
24- }  
25-
```

```
class Main {  
    public static void tea(){  
        System.out.println("-----Tea-----");  
        System.out.println("Boil water");  
        System.out.println("Put tea bag in cup");  
        System.out.println("Pour hot water");  
        System.out.println("Steep 2-5 mins");  
        System.out.println("Remove bag, add sugar/milk if you like");  
    }  
  
    public static void coffee(){  
        System.out.println("-----Cofee-----");  
        System.out.println("Boil water");  
        System.out.println("Add 1-2 tsp instant coffee to cup");  
        System.out.println("Pour hot water");  
        System.out.println("Stir");  
        System.out.println("Add sugar/milk if you like");  
    }  
  
    public static void iceCofee(){  
        System.out.println("-----Ice Cofee-----");  
        System.out.println("Make strong hot coffee");  
        System.out.println("Let it cool (or use ice)");  
        System.out.println("Fill glass with ice");  
        System.out.println("Pour coffee over ice");  
    }  
}
```

```
System.out.println("Add sugar/milk if you like");  
}  
  
public static void main(String[] args) {  
    System.out.println("you want tea");  
    tea();  
  
    System.out.println("-----");  
  
    System.out.println("you want coffee");  
    coffee();  
  
    System.out.println("-----");  
  
    System.out.println("you want icecoffee");  
    iceCoffee();  
  
}
```

you want tea

-----Tea-----

Boil water

Put tea bag in cup

Pour hot water

Steep 2–5 mins

Remove bag, add sugar/milk if you like

you want coffee

-----Coffee-----

Boil water

Add 1–2 tsp instant coffee to cup

Pour hot water

Stir

Add sugar/milk if you like

you want icecoffee

-----Ice Coffee-----

Make strong hot coffee

Let it cool (or use ice)

Fill glass with ice

Pour coffee over ice
Add sugar/milk if you like

```
class Main {  
    public static void printOneToTen() {  
        // 1 - 10  
        for(int i = 1; i<=10; i++) {  
            System.out.println(i);  
        }  
    }  
  
    public static void printOneToHundred() {  
        // 1 - 100  
        for(int i = 1; i<=100; i++) {  
            System.out.println(i);  
        }  
    }  
  
    public static void main(String[] args) {  
        printOneToTen();  
        printOneToTen();  
        printOneToHundred();  
  
    }  
}  
1  
2  
3  
4  
5
```

6
7
8
9
10
1
2
3
4
5
6
7
8
9
10
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29

30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73

74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

Types of Java Methods

Predefined Methods

User-defined Methods



◆ Parameterized Methods

Java run wenne line by line habai method ekak athule,
Mulinma run wanna patan ganne main method eken, api method eka call
karanawa main method ekata ekath run wenne eka athule line by line, num1
num2 variable hadala thiyanne block ekak athule nisa , pita block eka mehema
num1, num2 kila variable thinawada kyala eya danne na(variable scope case
eka)
Api me values sambanda karaganna widihak thinawa,
**Wena method ekaka thiyena values sambanda karaganna ona awasthawalata
thamai parameterized method kiyanne**

Mema wadeta api statement dekak yodagannawa,

- Aththatama wenne eka method ekaka thiyena argument values eliyata danakota thawath method ekaka thiyena parmiter valin allagena variable ekakata assign karagena ema method eka thula eya baawithaa karana eka

◆ Parameter Statement (Parameters)

Parameter එක කියන්නේ mokadda?

Parameters kiyanne method ekakata pass karana **values** walata place holders. Me values method eka define karankota thamai denna.

Example from your code:

```
public static void findTotal(int number1, int number2) {
```

මෙහි **number1** සහ **number2** කියන්නේ parameters.

මේවා method එකට යැවෙන්න යන අගයන් තාවකාලිකව represent කරනවා.

◆ Argument Statement (Arguments)

Argument කියන්නේ mokadda?

Arguments kiyanne actual values (literal values or variables) method ekakata yawanna denna values. Me values method ekak call karanakota denna.

 Example from your code:

```
int num1 = 45;  
int num2 = 56;  
findTotal(num1, num2); // <-- argument statement
```

මෙහි **num1** සහ **num2** කියන්නේ arguments. ඒ කියන්නේ actual values (45, 56) method එකට යවනවා

```
class Main{  
    public static void finaTotal() {  
        int total = num1 + num2;  
        System.out.println("Total IS : " + total);  
    }  
  
    public static void main(String[] args) {  
        int num1 = 45;  
        int num2 = 56;  
        finaTotal();  
    }  
}
```

Error

Hari gassanawidiha,

```
class Main{
```

```

public static void finaTotal(int num1, int num2) {
    int total = num1 + num2;
    System.out.println("Total IS : " + total);
}

public static void main(String[] args) {
    int num1 = 45;
    int num2 = 56;
    finaTotal();
}
}

```

error

```

class Main{
    public static void finaTotal(int num1, int num2) {
//parameters
        int total = num1 + num2;
        System.out.println("Total IS : " + total);
    }

    public static void main(String[] args) {
        int num1 = 45;
        int num2 = 56;
        finaTotal(num1, num2); //Argument Statement
    }
}

```

Total IS : 101

Num1 value eka number1 kiyana parameter ekata join wenawa ,Num2 value eka number2 kiyana parameter ekata join wenawa

```

class Main{
    public static void findTotal(int number1, int number2){//paramiters

        System.out.println("number1 : " + number1);
        System.out.println("number2 : " + number2);

        int total = number1 + number2;
        System.out.println("Total is : " + total);
    }

    public static void main(String[] args) {
        int num1 = 45;
        int num2 = 56;
        findTotal(num1, num2); //Argument Statement
    }
}

```

```
}
```

number1 : 45
number2 : 56
Total is : 101

```
class Main{  
    public static void findTotal(int number1, int  
number2) //paramiters  
  
        System.out.println("number1 : " + number1);  
        System.out.println("number2 : " + number2);  
  
        int total = number1 + number2;  
        System.out.println("Total is : " + total);  
}  
  
public static void get() {  
    int num1 = 45;  
    int num2 = 56;  
    findTotal(num1, num2); //Argument Statement  
}  
  
public static void main(String[] args) {  
    get();  
}  
}
```

number1 : 45
number2 : 56
Total is : 101

```
class Main{  
    public static void finaTotal(int number1, int number2, int  
number3) //paramiters  
  
        System.out.println("number1 : " + number1);  
        System.out.println("number2 : " + number2);  
        System.out.println("number2 : " + number3);  
  
        int total = number1 + number2;  
        System.out.println("Total is : " + total);  
}
```

```
}

public static void main(String[] args) {
    int num1 = 45;
    int num2 = 56;
    int num3 = 30;
    finaTotal(num1, num2, num3); //Argument Statement
}
}

number1 : 45
number2 : 56
number2 : 30
Total is : 101
```

```
1 * class Main {
2 *     public static void findTotal(int num1, int num2) {
3 *         int total = num1 + num2;
4 *         System.out.println("Total = " + total);
5 *     }
6 *
7 *     public static void main(String[] args) {
8 *         findTotal(1, 2);
9 *         findTotal(3, 2);
10 *        findTotal(10, 20);
11 *    }
12 * }
13 *
```

```
import java.util.Scanner;

class Main {

    public static void findMax(int number1, int number2, int
number3) { //paramiters
        int max = number1;
        if(number2 > max) {
            max = number2;
        }
        if(number3 > max) {
            max = number3;
        }
        System.out.println("Max: " + max);
    }

    public static void findMin(int number1, int number2, int
number3) {
        int min = number1;
        if(number2 < min) {
            min = number2;
        }
        if(number3 < min) {
            min = number3;
        }
        System.out.println("Min: " + min);
    }

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Input Number 01: ");
        int num1 = input.nextInt();

        System.out.print("Input Number 02: ");
        int num2 = input.nextInt();

        System.out.print("Input Number 03: ");
        int num3 = input.nextInt();

        findMax(num1, num2, num3); //Argument Statement
        findMin(num1, num2, num3); //Argument Statement
    }
}
```

Input Number 01: 12

Input Number 02: 23

Input Number 03: 34

Max: 34

Min: 12

Volume = parimawa

```
class Main {  
  
    public static void findVolume(int l, int w, int h) {  
        int volume = l * w * h;  
        System.out.println("Volume: " + volume);  
    }  
  
    public static void main(String[] args) {  
        int length = 10;  
        int width = 20;  
        int height = 30;  
  
        findVolume(length, width, height);  
    }  
}
```

Volume: 6000

```
import java.util.Scanner;  
class Main {  
    public static void findVolume(int l, int w, int h) {  
  
        int volume = l * w * h;  
        System.out.println("Volume: " + volume);  
    }  
}
```

```

public static void main(String[] args) {
    Scanner input = new Scanner(System.in);
    System.out.print("Enter length :");
    int length = input.nextInt();

    System.out.print("enter width :");
    int width = input.nextInt();

    System.out.print("enter height :");
    int height = input.nextInt();

    findVolume(length, width, height);
}
}

```

Enter length :23

enter width :43

enter height :232

Volume: 229448

◆ Return in methods

eke aththatama wenne

Return eken aththatama wenne ,**Return kiyana s statement eka thiyyena method eke eya kara kara hitapu wada tika iwara karanawa,**
Return eka thiyyena(findTotal) method eken eliyata enawa, value ekak thinawanam call karapu thanata yawala wada nawaththanawa.

Return karana data type eka hadunala na ekai error enne, eka kiyanne return throw void eka adala data type eka widihata ena karala

```

class Main{
    public static void finaTotal(int number1, int number2){
        int total = number1 + number2;
        return total;
    }

    public static void main(String[] args) {
        System.out.println(finaTotal(5, 6));
    }
}

```

error

Return karana data type eka hadunala na ekai error enne, eka kiyanne return throw void eka adala data type eka widihata ena karala

```
class Main{  
    public static int finaTotal(int number1, int number2){  
        int total = number1 + number2;  
        return total;  
    }  
  
    public static void main(String[] args) {  
  
        System.out.println(finaTotal(5,6)); //11  
        // System.out.println(11); // meka nikan menna me  
wage samnya print wena de thamai uda sop ekenuth wenne  
    }  
}
```

11

Return karana eka findTotal(5,6)n allagannawa

Int sum 5 + 6 sop ekak haraha print karanawa wage, methanadi wenne, sub method eke ena calculation eka iwara unama eya kinawa total eka return yawanna kila ethanadi wenne total eka hoyagena eka call karapu thanata value eka yawala method eke eya kara kara hitapu wade iwara karanawa eka, ethanata gihilla athana thiyyena total eka vol kiyana variable name ekata assign karanawa assign karala ema value eka sop(vol) adarayen print karanawa

```
class Main {  
    public static int findTotal(int number1, int number2) {  
        int total = number1 + number2;  
        return total;  
    }  
  
    public static void main(String[] args) {  
        // int sum = 5 + 6;  
        // System.out.println(sum);  
        int vol = findTotal(5,6);  
        System.out.println(vol);  
    }  
}
```

11

Samanyayen return ekak main method ekata recall wenakota api ekata variable ekak hadala e yatathe thamai danne

```
class Main {
    public static int findVolume(int l, int w, int h) {
        int volume = l * w * h;
        return volume;
    }

    public static double findCapacity(int vol) {
        double capacity = (double)vol/1000;
        return capacity;// double ekak nam return wenne void
wenuwata double danna wenawa
    }

    public static void main(String[] args) {
        int length = 100;
        int width = 50;
        int height = 20;

        // volume
        int volume = findVolume(length, width, height);
        System.out.println("Volume: " + volume);

        // capacity
```

```
        double capacity = findCapacity(volume);
        System.out.println("Capacity: " + capacity);

    }
}
```

Volume: 100000

Capacity: 100.0

```
import java.util.Scanner;

class Main {
    public static int findVolume(int l, int w, int h) {
        int volume = l * w * h;
        return volume;
    }

    public static double findCapacity(int volume) {
        double capacity = (double)volume/1000;
        return capacity;// return karanne double data type
ekaka value ekak nisa void wenuwata double danna ona
    }
}

public static void main(String[] args) {
    Scanner input = new Scanner(System.in);
    System.out.print("Enter length: ");
    int length = input.nextInt();
    System.out.print("Enter width: ");
    int width = input.nextInt();
```

```

        System.out.print("Enter height: ");
        int height = input.nextInt();

        int vol = findVolume(length,width,height);
        System.out.println("Volume = " + vol);
        double capacity = findCapacity(vol);
        System.out.println("Capacity = " + capacity);
    }
}

```

Enter length: 2

Enter width: 34

Enter height: 45

Volume = 3060

Capacity = 3.06

◆ Recursive Method

api call karana method eka athulema apahu e call karana method ekama call karana method ekama call karana eka

A **recursive method** is a method that **calls itself** to solve a problem **step by step**, usually by reducing the size of the problem

```

class Example{
    public static void myMethod(){
        System.out.println("Start myMethod");
        myMethod();
        System.out.println("End myMethod");
    }
    public static void main(String args[]){
        myMethod();
    }
}

```

```
class Example{
    public static void myMethod(int i){
        System.out.println("i : " + i);
        myMethod(++i);
    }
    public static void main(String args[]) {
        myMethod(0);
    }
}
```

```
class Example{
    public static void myMethod(int i){
        System.out.println("i : " + i);
        if(i < 10)
            myMethod(++i);
    }
    public static void main(String args[]) {
        myMethod(0);
    }
}
```

i:0
i:1
i:2
i:3
i:4
i:5
i:6
i:7
i:8
i:9
i:10

◆ Java Memory

```
class Main {  
    public static void sing() {  
        System.out.println("Sing!");  
    }  
  
    public static void dance() {  
        System.out.println("Dance!");  
    }  
  
    public static void run() {  
        System.out.println("Run!");  
    }  
  
    public static void main(String[] args) {  
        System.out.println("Hello 1");  
        sing();  
        System.out.println("Hello 2");  
        run();  
        System.out.println("Hello 3");  
        dance();  
    }  
}
```

```
}
```

```
Hello 1
```

```
Sing!
```

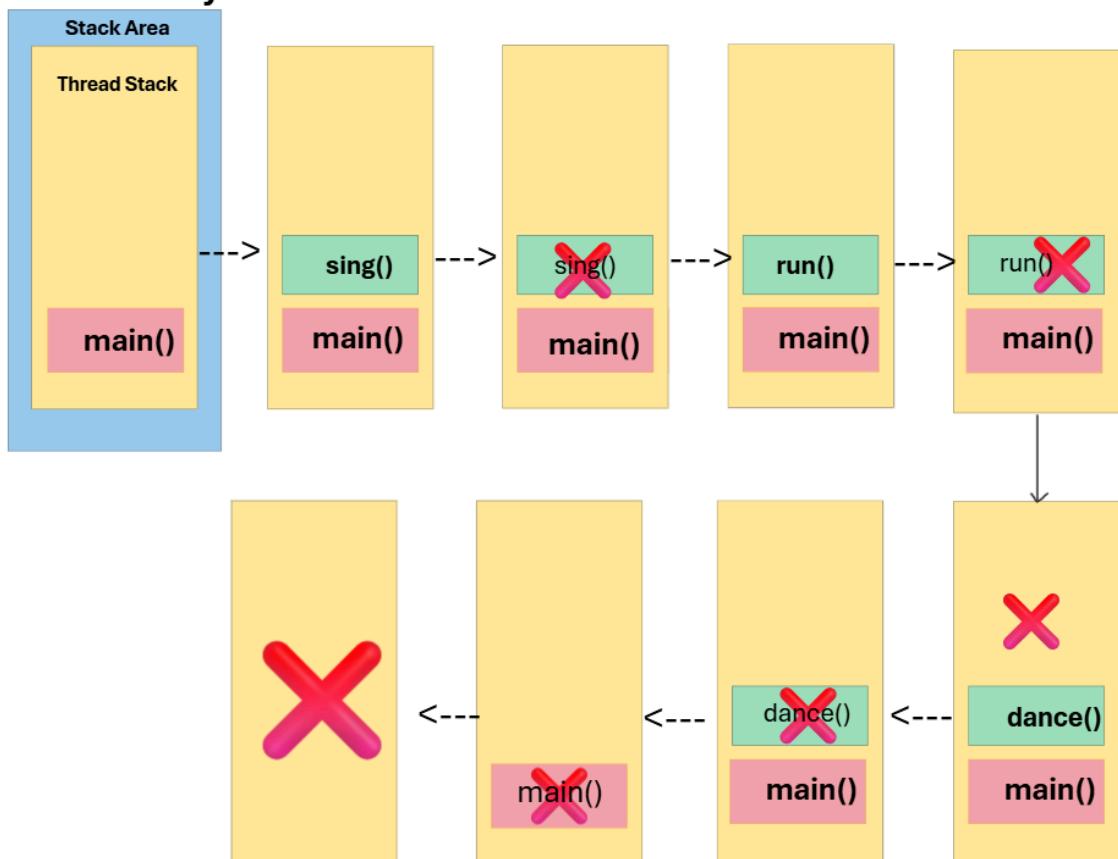
```
Hello 2
```

```
Run!
```

```
Hello 3
```

```
Dance!
```

Java memory



```
class Main {  
  
    public static void run() {  
        System.out.println("Run! ");  
        jump();  
    }  
  
    public static void jump() {  
        System.out.println("jump! ");  
    }  
  
    public static void main(String[] args) {  
    }  
}
```

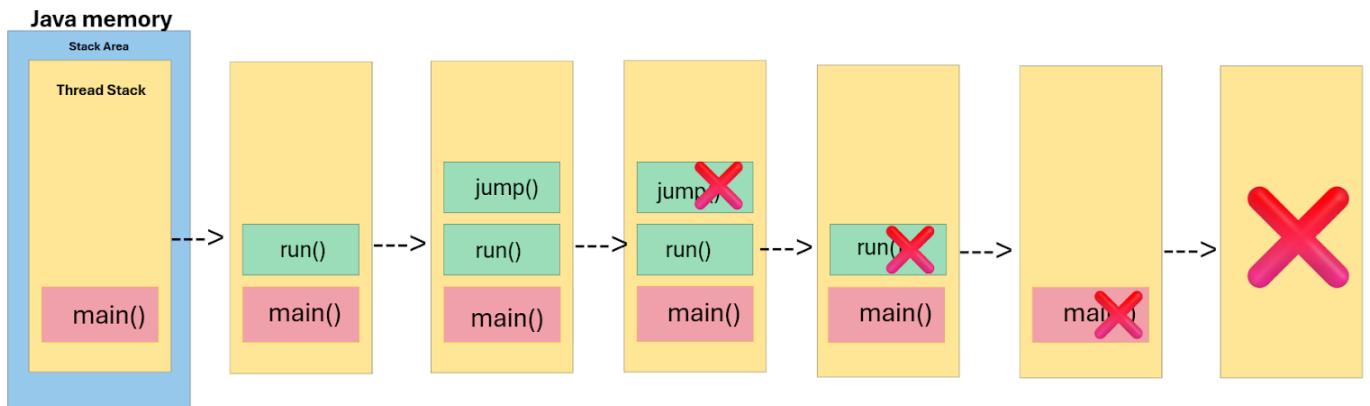
```

        System.out.println("Hello");
        run();

    }

}

```



garbage collector collector eken thamai iwath karanne unusable stack frames (methods)

```

class Main {
    public static void sing() {
        System.out.println("Sing!");
    }

    public static void dance() {
        System.out.println("Dance!");
    }

    public static void run() {
        System.out.println("Run!");
        jump();
    }

    public static void jump() {
        System.out.println("jump!");
    }

    public static void main(String[] args) {
        System.out.println("Hello 1");
        sing();
        System.out.println("Hello 2");
        run();
        System.out.println("Hello 3");
        dance();
    }
}

```

```
    }
}

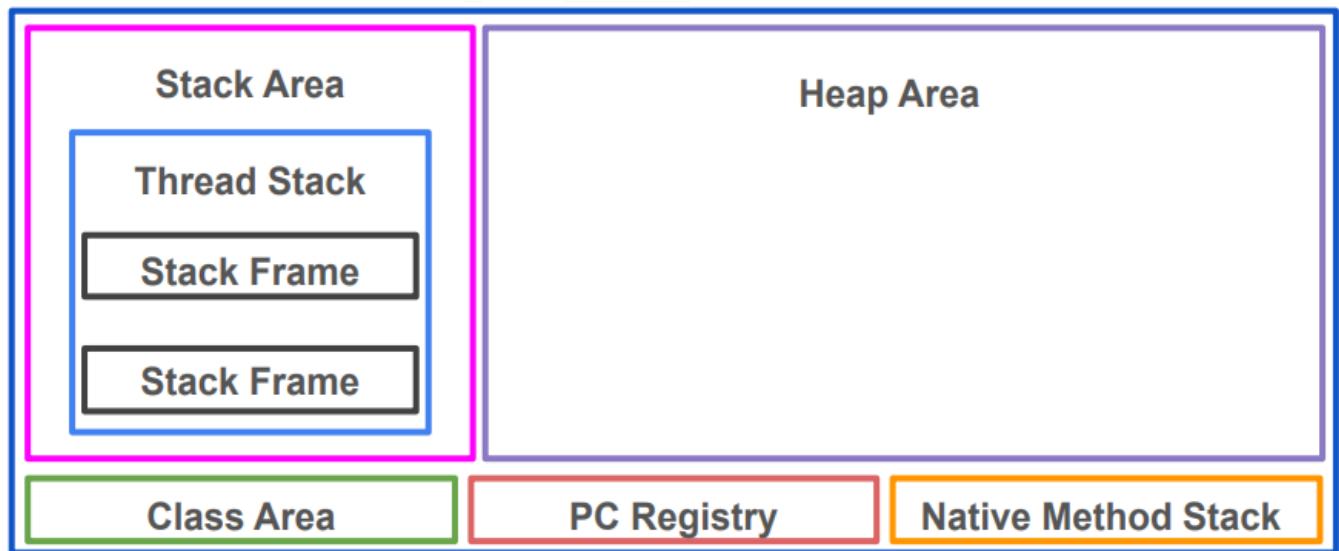
Hello 1
Sing!
Hello 2
Run!
jump!
Hello 3
Dance!
```

```
class Main {
    public static void sing() {
        String a = "Hello";
        double b = 1.5;
    }

    public static void main(String[] args) {
        int x = 100;
        int y = 200;
        boolean z = false;
        sing();
    }
}
```

garbage collector collector eken thamai iwath karanne unusable stack frames
(methods)

•JVM Memory



Types of Java Methods

Predefined Methods

User-defined Methods



◆ Predefined methods,

Java standers Library samaga inbuild paminena methoda

Ex: random

◆ Random (ahambu)

- import java.lang.Math;

මෙය කියන්නේ Math class එක import කරන එක.

java.lang කියන package එක දැනටමත් Java වල default එකක් නීසා මෙය import කරන්නන් අවශ්‍ය නෑ. එහෙමයි කියන්නේ මෙක ලිඛුණන් ලියන්නේ නැත්තන් වැඩිහිටි

- System.out.println(Math.random());

මෙය Math.random() method එකෙන් 0.0 සහ 1.0 අතර එකක් random decimal number (අභ්‍යු දැගම අංකයක්) generate කරලා print කරනවා.

`Math.random()` → double වර්ගයේ අගයක් return කරයි.

මෙය සුම වරෙකම $0.0 \leq \text{value} < 1.0$ ලෙසින් එයි.

`Math.random() => $0.0 \leq \text{value} < 1.0$`

`Math.random()` -> 0.0 ho 0.0- 1.0 th athara (double)

```
import java.lang.Math;

class Main {
    public static void main(String[] args) {
        System.out.println(Math.random()); // 0.0 - 1.0
    }
}
```

```
import java.lang.Math;

class Main {
    public static void main(String[] args) {
        System.out.println(Math.random()); // 0 - 9
        double result= Math.random();
        System.out.println(result);
    }
}
```

Variable ekak ha sop eka thula dewidiyatama random values use karanna puluwan

Methana math import kare nathath wada , ewunata math import karalama liyaa eka thamai best practices wenne

```
(int)Math.random() * 10) -> saama warakama 0,
Mokaa 0-1 thar 0.00wlues wala integer value eka 0 nisa
import java.lang.Math;

class Main {
    public static void main(String[] args) {
        System.out.println((int)Math.random()); // 0
    }
}
```

මෙය return කරන එකක්:

Math.random() => $0 \leq \text{value} < 1$
Mema values wala int value eka hambawenawa

මෙහිදී **(int)Math.random()** එකෙන් **decimal (double)** අයයේ wenuwata ema desimal (double) value eke integer(puurna agaya).value eka hamba wenawa

```
(int)0.234832 => 0
(int)0.998324 => 0
(int)0.00001   => 0
```

අනිවාර්යයෙන්ම මූලික "int" අයය 0 විතරයි ලැබෙන්නේ.

Math.random() defoult enne decimal value ekak nisa api eyawa me widihata doble ekak int karanna gihilla hariyanne na, apita ekata thawa duratath Math.random kiyanne int ekak kila bala karanna wenawa

```
import java.lang.Math;
class Main{
    public static void main(String args[]) {

        int random = Math.random();
        System.out.println(random);
    }
}
```

error

```
import java.lang.Math;
class Main{
```

```

public static void main(String args[]) {
    int random = (int) Math.random() * 10;
    System.out.println(random);
}

```

0

(int)(Math.random() * 10) -> 0 ho 0-9(int)

```

import java.lang.Math;

class Main {
    public static void main(String[] args) {
        System.out.println((int) (Math.random() * 10)); // 0
- 9
    }
}

```

Mehema int kila danme nathnam apata default denne double value ekak

$\text{Math.random()} * 6 // \Rightarrow 0*10 \leq \text{value} < 1*10$
 $\Rightarrow 0 \leq \text{value} < 10$
 $\Rightarrow 0, 1, 2, 3, 4, 5, 6, 7, 8, 9$

90 -100 -> $0*11+90 \leq \text{value} < 1*11+90$

Variable ekak dalath liyanna puluwan

```

import java.lang.Math;
class Main {
    public static void main(String[] args) {
        int result = (int) (Math.random() * 10); // 0.0 - 9.0
        System.out.println(result);
    }
}

```

(int)(Math.random() * 6) + 1 -> 0 ho 0 - 7 ahara sankya

```

import java.lang.Math;

```

```

class Main {
    public static void main(String[] args) {
        System.out.println((int)(Math.random()*6) + 1); // 1 - 6
    }
}

```

Math.random() // => $0.0 \leq \text{value} < 1.0$

```

import java.lang.Math;

class Main {

    public static void main(String[] args) {

        int result = (int)(Math.random() * 6)+1; // 1 - 6

        System.out.println(result);
    }
}

```

මෙම අගය * 6 කරනවා:

$\text{Math.random()} * 6 // \Rightarrow 0.0 * 6 \leq \text{value} < 1.0 * 6$
 $\Rightarrow 0.0 \leq \text{value} < 6.0$
 $\Rightarrow 0, 1, 2, 3, 4, 5$

අදාළ නොවන:

- $0.0 * 6 = 0.0$
- $0.999 * 6 \approx 5.994$

$(\text{int})(\text{Math.random()} * 6) // \Rightarrow 0.0 \leq \text{value} < 6.0$
 $\Rightarrow 0, 1, 2, 3, 4, 5$

එම අගයට + 1 එකතු කරනවා:

```
(int)(Math.random() * 6) + 1 // => 0.0+1 ≤ value < 6.0+1  
= > 1.0 ≤ value < 7.0  
=> 1, 2, 3, 4, 5, 6
```

```
import java.lang.Math;  
  
class Main {  
    public static void main(String[] args) {  
        int result = (int)(Math.random()*6) + 1;  
        System.out.println(result); // 1 - 6  
    }  
}
```

අන්තටම ඔයා දිපු මේ Java code දෙකේ නියෙන්නේ ලොකු වෙනසක් නෑ — **output** එකමයි. හැඳුනී **code structure** එකේ තුව සූල වෙනසක් නියෙනවා👉

◆ පළවෙනි Code එක:

```
System.out.println((int)(Math.random()*6) + 1);
```

👉 මෙකකදී **Math.random()** call එක **directly print** කරනවා.

අගය **store** නොවේයි. (temporary calculation)

◆ දෙවෙනි Code එක:

```
int result = (int)(Math.random()*6) + 1;  
System.out.println(result);
```

👉 මෙකකදී **Math.random()** call එකක් ලැබෙන අගය **result** කියන **variable** එකකා **store** කරලා නියෙනවා.

ඊට පස්සේ එක **print** කරනවා

- 👉 දෙකමම output එක එකමයි.
 - 👉 Variable එකක් භාවිතා කිරීම ඔබට code එක clear කරලා, value එක නැවත භාවිතා කරන්න උදෑසී කරයි.
 - 👉 එම නිසා දෙවෙන් විධිය coding වලදී **best practice** එකක්.

Random Numbers

||||| 222

```
import java.util.Random;
import java.util.Scanner;

class Example{
    public static void main(String[] args) {
        // Scanner input = new Scanner(System.in);
        // System.out.print("Input number : ");

        Random input = new Random();

        int num = input.nextInt();
        System.out.println("Your Number is : " + num);
    }
}
```

223

```
import java.util.Random;

class Example{
    public static void main(String[] args) {
        Random input = new Random();

        for(int i = 0; i < 20; i++){
            int num = input.nextInt(); // min to max int
            System.out.println(num);
        }
    }
}
```

224

```
import java.util.Random;

class Example{
    public static void main(String[] args) {
        Random input = new Random();

        for(int i = 0; i < 20; i++){
            int num = input.nextInt(10); // 0 to 9
            System.out.println(num);
        }
    }
}

=====
```

225

```
import java.util.Random;

class Example{
    public static void main(String[] args) {
        Random input = new Random();

        for(int i = 0; i < 20; i++){
            int num = input.nextInt(10) + 1; // 1 to 10
            System.out.println(num);
        }
    }
}
```

```
=====
```

226 Exercise, print 20 random numbers between 50 to 100

```
=====
```

```
import java.util.Random;

class Example{
    public static void main(String[] args) {
        Random input = new Random();

        for(int i = 0; i < 20; i++){
            int num = input.nextInt(51) + 50;
            System.out.println(num);
        }
    }
}
```

```
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
```

227 Example 1

```
=====
```

```
import java.util.Random;

class Example{
    public static void main(String[] args) {
        Random input = new Random();

        for(int i = 0; i < 100; i++){
            int num = input.nextInt(101);
            System.out.println(num);
        }
    }
}
```

```
||||||||||||||||||||||||||||||||||||||||||||||||||||
```

228 Example 2

```
=====
```

```
import java.util.Random;

class Example{
    public static void main(String[] args) {
        Random input = new Random();

        int num = 0;
        while(num != 100){
            num = input.nextInt(101);
            System.out.println(num);
        }
    }
}
```

length()

```
class Main{
    public static void main(String args[]){
        String x = "java";
        System.out.println(x.length());
    }
}
```

4

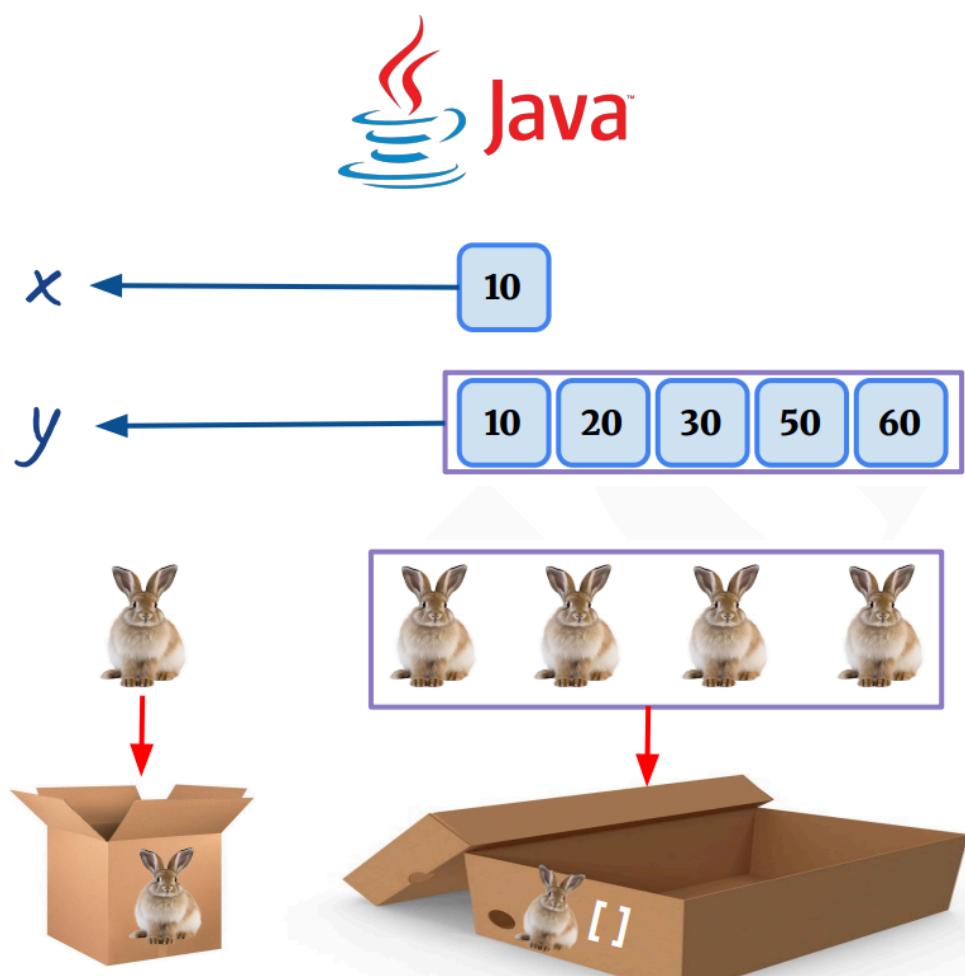
```
class Main{
    public static void main(String args[]){
        int x = 574;
        System.out.println(x.length());
    }
}
```

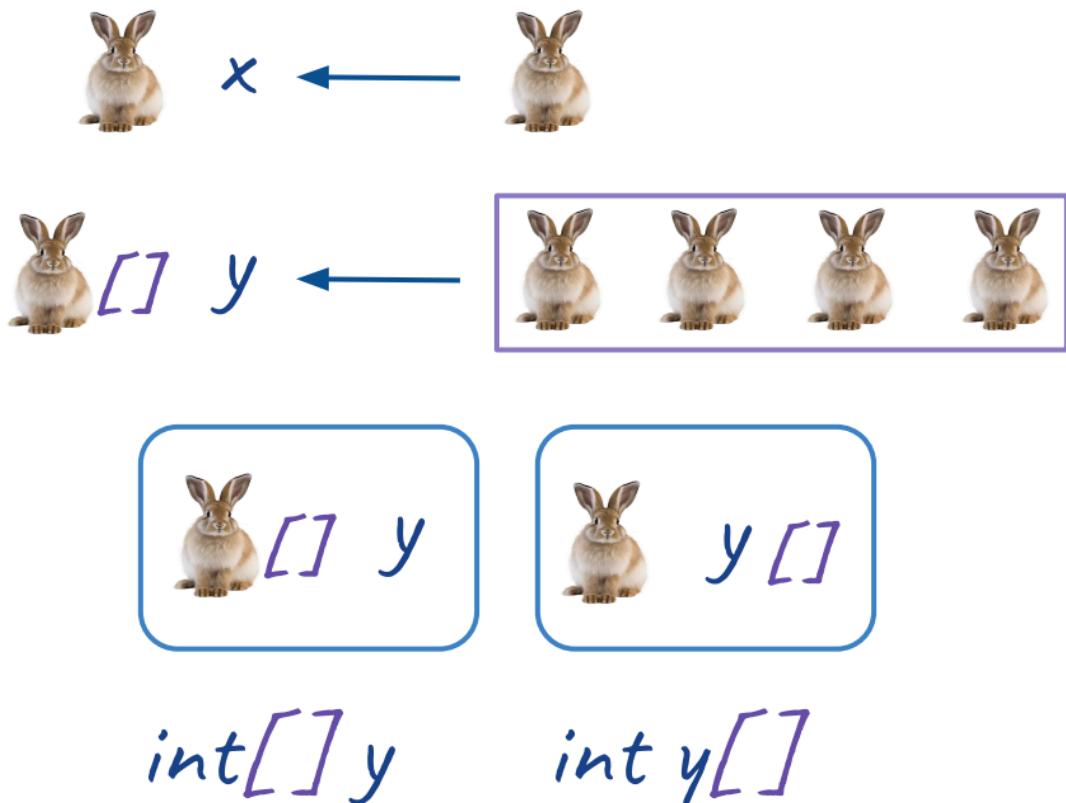
WRONG

◆ Java Arrays

Multiple value store at single variable

Arrays in Java are one of the most fundamental data structures that **allow us to store multiple values of the same type in a single variable.**





● Java Array කියන්නේ මොකක්ද?

Java array එකක් කියන්නේ එකම දත්ත වර්ගයක (same data type) අගයන් කිහිපයක් එකම නාමයක් යටතේ ගබඩා කරගන්න කුමයක්.

Arrays - Non primitive

What is element and index ?

```
0 , 1 , 2 , 3 , 4
int[] numbers = {10, 20, 30, 40, 50};
<----->
          Array Length
```

මෙහි:

10 → 1වෙති element (index 0)

20 → 2වෙති element (index 1)

30 → 3වෙති element (index 2)

40 → 4වෙති element (index 3)

50 → 5^{වෙනු} element (index 4)

- **Last index = Array Length - 1**

Array Length = List index +1

numbers[0] ->10
numbers[1] ->20
numbers[2] ->30
numbers[3] ->40
numbers[4] ->50
numbers[5] -> error

numbers.length ->5

Numbers array eke awayawa ganana

Numbers[numbers.length-1] -> numbers[4] -> 50

Numbers array eke last index eka (anthima awayawaya)

```
import java.util.*;
class Main{
    public static void main(String args[]) {
        //      element      1   2   3   4   5   6   7   8   9   ....
        //      index       0   1   2   3   4   5   6   7   8   ....
        int []x ={10, 20, 30, 40, 50, 8554, 7565 };

        System.out.println(x[0]);
        System.out.println(x[1]);
        System.out.println(x[2]);

        System.out.println(x.length);
        System.out.println(x.length);
        System.out.println(x[x.length-1]);
        System.out.println(x[x.length/2]);

    }
}
```

int[] a

Int[] - Array type

a - Array name

Arrays use karnakota api arrays import karaganna ona api ekko arrays import karanawa nathnam all (*) import karanawa

Import java.util.Arrays; util file eke Array package eka genwanawa

Import java.util.*; util file eke serama packages genawnawa

● උදාහරණයක්:

```
int[] numbers = new int[5];
```

මෙකෙන් කියන්නේ int (ඉන්ටිජර්) වර්ගයේ අංක 5ක් තැන්පත් කරන්න array එකක් සාදනවා කියන එකයි.

● Array එකක් සංජුවම නිර්මාණය කිරීම:

```
int[] marks = {75, 80, 90, 85, 95};
```

මෙහි marks කියන array එකට අගයන් 5ක් දාලා නියෙනවා.

● Array එකේ අංග වලට පිවිසීම (Access):

```
System.out.println(marks[0]);
```

මෙකෙන් පලමු අගය (index 0) print කරනවා. මෙතන index 0 නමට අදාළව 75 print වේ.

● Array එකේ අගයන් වෙනස් කිරීම:

```
marks[2] = 100;
```

මෙකෙන් තෙවැනි අගය (index 2) 90 → 100 ලෙස වෙනස් වේ.

● Array එක loop එකකින් පරිච්ඡේද කිරීම (Looping through):

```
for (int i = 0; i < marks.length; i++) {  
    System.out.println(marks[i]);  
}
```

හෙයා:

```
for (int mark : marks) {  
    System.out.println(mark);  
}
```

● පූර්ණ උදාහරණයක්:

```
public class Main {  
    public static void main(String[] args) {  
        String[] fruits = {"Apple", "Banana", "Mango"};  
  
        for (int i = 0; i < fruits.length; i++) {  
            System.out.println(fruits[i]);  
        }  
    }  
}
```

Output:

```
Apple  
Banana  
Mango
```

 Array එක ගැන මතක තබාගන්න වැදගත් දේවල්:

විශේෂාංග ය

විස්තරය

 Index	0 සිට පටන් ගන්නවා
 Size	Array එකක් සාදුද්දී එහි විශාලත්වය පෙර නිශ්චිත කළ යුතුයි
 Data type	එක array එකක් සියලුම අගයන් එකම වර්ගයක විය යුතුයි (int, String, etc.)

Arrays are objects in Java  → According to the Java Language Specification, every array in Java is an object, regardless of its type.

- Java walin `int[] numbers = new int[5];` kiyala array ekak haduwama, eka **object ekak** heap memory eke hadenawa.
 - Array ekak hadanna `new` keyword eka use karanna wenawa (new object hadanna wage).
-

Java Array Creation

10.3. Array Creation

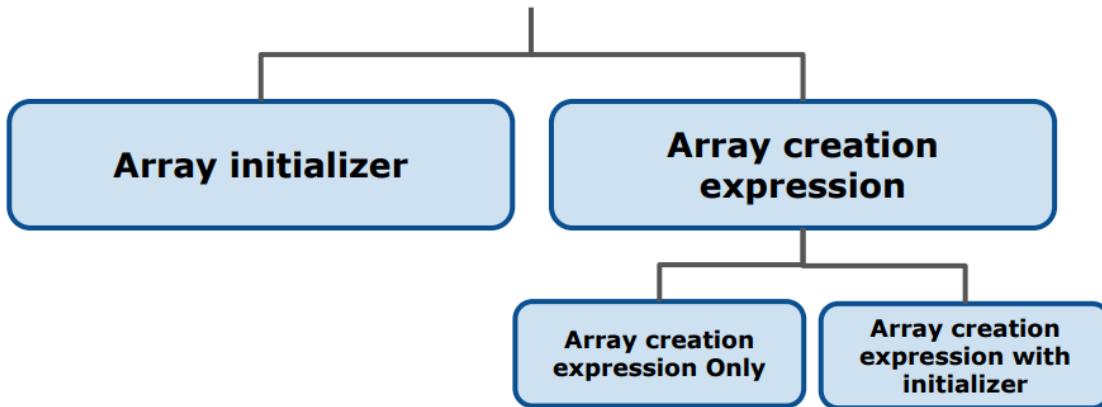
An array is created by an array creation expression ([\\$15.10.1](#)) or an array initializer ([\\$10.6](#)).

An array creation expression specifies the element type, the number of levels of nested arrays, and the length of the array for at least one of the levels of nesting. The array's length is available as a final instance variable `length`.

An array initializer creates an array and provides initial values for all its components.

(<https://docs.oracle.com/javase/specs/jls/se21/html/jls-10.html#jls-10.6>)

◆ Java Array Creation



◆ **Array Initializer**, awasthawa

This is a shortcut way to declare and initialize an array, often called **shorthand initialization**, but the **official term is Array Initializer**.

```
int[] y = {1, 2, 3, 4, 5};  
String[] colors = {"Red", "Green", "Blue"};
```

```
class Main {  
    public static void main(String[] args) {  
        // [10, 20, 30, 40, 50]  
        int[] a = {10, 20, 30, 40, 50};  
  
        // ["Hello", "Hi", "WOW"]  
        String[] b = {"Hello", "Hi", "WOW"};  
  
        // [true, false, false, true, true, false]  
        boolean[] c = {true, false, false, true, true, false};  
  
        // ['A', 'B', 'C', 'D']  
        char[] d = {'A', 'B', 'C', 'D'};  
  
        // [0.1, 0.2, 5.1, 4.5]  
        double[] e = {0.1, 0.2, 5.1, 4.5};  
    }  
}
```

Mata ona arrays tika print karanna komada eka karanne,

Mema karama print wenne memory address eka,

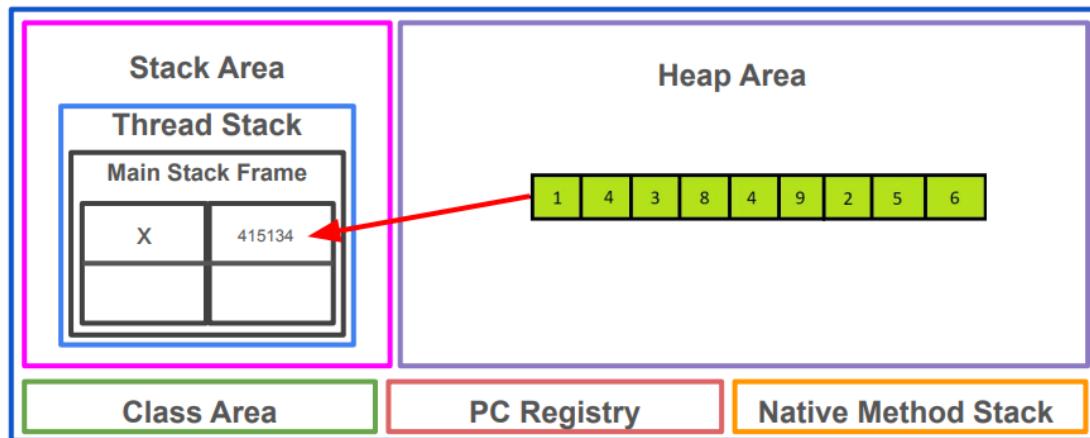
```
class Main {  
    public static void main(String[] args) {  
        int x = 10;  
        int[] a = {10, 20, 30, 40, 50};  
        System.out.println(a); // print memory address  
    }  
}
```

[I@6acbcfc0

Ai mehema wenne?

Array memory chart eka methnata danna

Java Array in Java Memory



Hariyata arrays list eka print karana idihā

```
import java.util.*;
class Main {
    public static void main(String[] args) {
        int x = 10;
        int[] a = {10, 20, 30, 40, 50};
        System.out.println(a); // print memory address
        System.out.println(Arrays.toString(a)); // [10, 20, 30,
40, 50]
    }
}
// [I@6acbcfc0
[10, 20, 30, 40, 50]
```

◆ **Arrays.toString()**

It is used to easily **print the contents of an array as a readable string.**

```
int[] numbers = {1, 2, 3, 4, 5};

System.out.println(numbers); // Prints something like: [I@6d06d69c (memory address)

System.out.println(Arrays.toString(numbers)); // Prints: [1, 2, 3, 4, 5]
```

Arrays.toString(a) කියන්නේ මොකක්ද?

Java වල **Arrays.toString(array)** කියන්නේ

→ **array** එකේ **elements** විකක් **ලස්සන format** එකකින් **String** එකක් **ලෙස දක්වන method** එකක්.

- උදාහරණය:

```
int[] a = {10, 20, 30, 40, 50};
System.out.println(a);
```

// ➡️ මෙය print කරන්නේ array එක් memory address එකක් වගේ
විදියකට: [I@27f674d

```
System.out.println(Arrays.toString(a));  
// ➡️ මෙය print කරන්නේ: [10, 20, 30, 40, 50]
```

⚠️ මොකද සාමාන්‍යයෙන් System.out.println(a) වලින්ම print
කරගන්න බැරිද?

ඔව්, array එකක් System.out.println() වලින් direct print කළාත්,
➡️ එය object එකක memory reference (address) වගේ එකක් print කරනවා.

අදා:

[I@5ca881b5

```
import java.util.*; // all classes import kranawa  
class Main {  
    public static void main(String[] args) {  
        int[] a = {10, 20, 30, 40, 50};  
        System.out.println(Arrays.toString(a)); // Arrays tika string  
ekak widihata print wenawa  
    }  
}
```

```
import java.util.*;  
class Main {  
    public static void main(String[] args) {  
        String[] s = {"Hello", "Hi", "WOW"};  
        System.out.println(s); // memory address  
        System.out.println(Arrays.toString(s)); // [Hello, Hi,  
WOW]  
    }  
}  
  
[Ljava.lang.String;@6acbcfc0  
[Hello, Hi, WOW]
```

```
import java.util.*;
class Main {
    public static void main(String[] args) {
        double[] d = {1.5, 2.5, 3.5, 4.5};
        System.out.println(d); // memory address
        System.out.println(Arrays.toString(d)); //
[1.5, 2.5, 3.5, 4.5]
    }
}
```

[D@6acbcfc0

[1.5, 2.5, 3.5, 4.5]

```
import java.util.*;
class Main {
    public static void main(String[] args) {
        boolean[] b = {true, false, true, false};
        System.out.println(b); // memory address
        System.out.println(Arrays.toString(b)); //
[true, false, true, false]
    }
}
```

[Z@6acbcfc0

[true, false, true, false]

```
import java.util.*;
class Main {
    public static void main(String[] args) {
        char[] c = {'A', 'B', 'C', 'D', 'E'};
        System.out.println(c); // memory address
        System.out.println(Arrays.toString(c)); //
    }
}
```

ABCDE

[A, B, C, D, E]

Character type wala nikanma SOP(c) danmath print wenawa value tika

```
import java.util.*;
class Main {
    public static void main(String[] args) {
        int[] a = {10, 20, 30, 40, 50};
        System.out.println(a.length); // 5
        String[] s = {"Hello", "Hi", "WOW"};
        System.out.println(s.length); // 3
    }
}
```

5

3

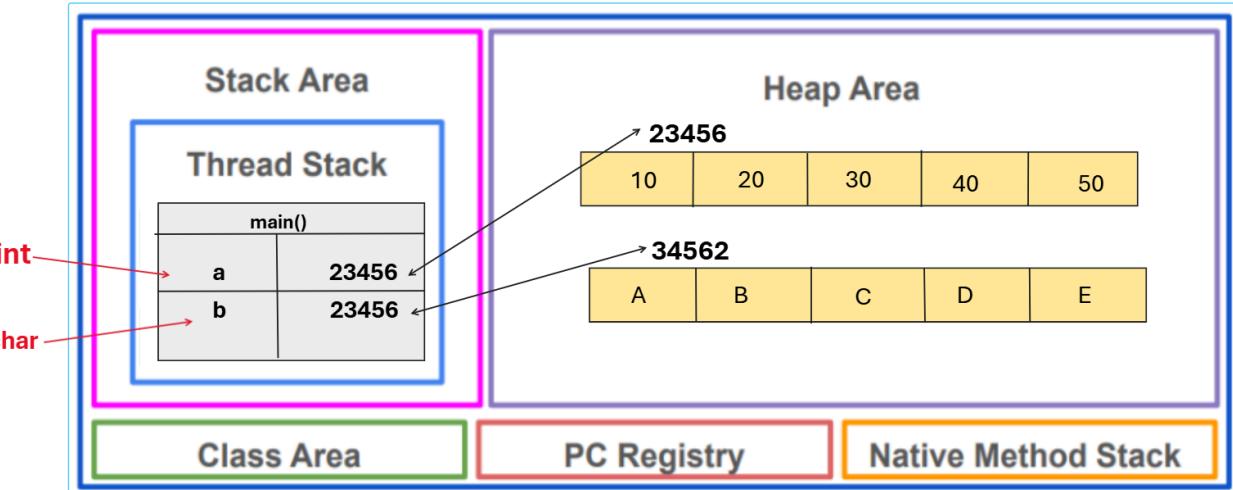
```
import java.util.*;
class Main {
    public static void main(String[] args) {
        int[] a = {10, 20, 30, 40, 50};
        System.out.println(Arrays.toString(a));
        String[] b = {"hello", "hi", "wow"};
    }
}
```

```
        System.out.println(Arrays.toString(b));
        double[] c = {1.5, 2.6, 7.5, 8.5};
        System.out.println(Arrays.toString(c));
        char[] d = {'A', 'B', 'C', 'D', 'E'};
        System.out.println(Arrays.toString(d));
        boolean[] e = {true, false, true, false, false};
        System.out.println(Arrays.toString(e));
    }
}

[10, 20, 30, 40, 50]
[hello, hi, wow]
[1.5, 2.6, 7.5, 8.5]
[A, B, C, D, E]
[true, false, true, false, false]
```

```
import java.util.*;
class Main {
    public static void main(String[] args) {
        int[] a = {10, 20, 30, 40, 50};
        System.out.println(a);
        String[] b = {"hello", "hi", "wow"};
        System.out.println(b);
        double[] c = {1.5, 2.6, 7.5, 8.5};
        System.out.println(c);
        char[] d = {'A', 'B', 'C', 'D', 'E'};
        System.out.println(d);
        boolean[] e = {true, false, true, false, false};
        System.out.println(e);
    }
}
```

```
[I@6acbcfc0
[Ljava.lang.String;@5f184fc6
[D@3feba861
ABCDE
[Z@5b480cf9
```



Apata mehema print karanna puluwan wenne api print karanne array group ekak nemei array eka athule print karana value ekak nisa, array group ekak print karanna ona nam `Array.toString()` danna wenwa

```
import java.util.*;
class Main {
    public static void main(String[] args) {
        int[] a = {10, 20, 30, 40, 50};
        System.out.println(a[2]); // 30
    }
}
```

◆ Length in Java Arrays

In Java, length is a special variable that tells you **how many elements are in an array.**

```
int[] numbers = {10, 20, 30, 40, 50};  
System.out.println(numbers.length); // Output: 5
```

```
class Main{  
    public static void main(String[] args){  
        String x = "HELLO";  
  
        System.out.println(x.length());  
    }  
}
```

Array use karane nathuwa variable ekaka length eka hoyana awasthawak

Array item kiyak thinawada kiyala balanne meken,

.length = Array awayawa ganana

```
[10, 20, 30, 40, 50, 60, 70]  
<----->
```

length = 7

```
import java.util.*;  
class Main {  
    public static void main(String[] args) {  
        int[] a = {10, 20, 30, 40, 50};  
        System.out.println(a.length); // 5  
  
        String[] b = {"hello", "hi", "wow"};  
        System.out.println(b.length); // 3  
  
        double[] c = {1.5, 2.6, 7.5, 8.5};  
        System.out.println(c.length); // 4
```

```

        char[] d = {'A', 'B', 'C', 'D', 'E'};
        System.out.println(d.length); // 5

        boolean[] e = {true, false, true, false, false};
        System.out.println(e.length); // 5
    }

}
5
3
4
5
5

```

```

import java.util.*;
class Main {
    public static void main(String[] args) {
        int[] a = {10, 20, 30, 40, 50, 76, 67, 65, 65, 5};
        System.out.println(a.length); //10
    }
}
10

```

Can You Change the Length of a Java Array?



No, once you create an array in Java, its length is fixed.

But is there a solution?

Yes! Let's find out in the upcoming slides.



◆ Array Creation Expression,

This is when you use the **new** keyword to directly create an array.

```
int[] x = new int[5];
String[] names = new String[3];
```

```
import java.util.*;
class Main {
    public static void main(String[] args) {
        int[] x = new int[5];
        // System.out.println(x.length); // 5
        // System.out.println(x); // memory address
        System.out.println(Arrays.toString(x)); // [0, 0, 0, 0, 0]
    }
}
[0, 0, 0, 0, 0]
```

```
int[] x = new int[5];
System.out.println(Arrays.toString(x)); // [0, 0, 0, 0, 0]
ഈക്കന് ആണേം:
```

അടി മേഖലയിൽ [0, 0, 0, 0] വരെ ഉണ്ടോ?

പിളിപ്പാർ:

Java ലൈംഗിക്ക്, array ലൈംഗിക്ക് new keyword ലൈംഗിക്ക് ഹാഡ്സ്ടീ, ലൈംഗിക്ക് elements default value ലൈംഗിക്ക് initialize ലൈംഗിക്ക് automatic ലൈംഗിക്ക്.

◆ മേഖലയിൽ code ലൈംഗിക്ക് ബലമുണ്ടോ:

```
int[] x = new int[5];
```

මෙතනින් කියන්නේ:

- x කියන int array එකක් සාදනවා
- එකේ විශාලත්වය (length) 5

new සාදන තීසා,

► Java default value දාලා තියෙනවා

Default Values



Data Type	Default Value
byte	0
short	0
int	0
long	0L
float	0.0f
double	0.0d
char	'\u0000' (null character)
boolean	false
Object References (e.g., String, arrays, custom objects)	null

◆ Java default values (array වර්ගයට අනුව):

Data Type	Default Value
int	0
double	0.0
boolean	false
char	\u0000 (null character)
String / Object	null

💡 එක තීසා:

int[] x = new int[5];

මෙහි x = [0, 0, 0, 0, 0]

→ එකඟී Arrays.toString(x) එකෙන් එන්නේ එ zero values 5ක්.

❖ තීගමනය:

- new int[5] කියන කොට array එක සාදාගන්නවා

- ඒකේ හරයටම කිසීම value set කරලා නැත්තම්,
▶ Java එක ඒක **default values** වලින් පිරවෙනවා
- **int** වල default value → 0

new

new කියන්නේ **අලුත් memory space** එකක් ගන්න එක

new arrays wala yodaganne emagin array එකක් හදනවා, values පස්සේ දාන්ත

● කොහොද **new** අනිවාර්ය වෙන්නේ?

- අනිවාර්යයි:

```
int[] x = new int[5];           // ✓
String[] names = new String[3]; // ✓
```

- අනිවාර්ය නෑ (අන්තටම shortcut එකක්):

```
int[] x = {10, 20, 30};        // ✓
```

මෙකෙදී **new** ඔයා ලියන්නේ නැත්තම්, Java compiler එක **අභ්‍යන්තරයෙන්ම (internally)** **new keyword** එක use කරනවා.

🔍 උදාහරණ 2ක පර්තය:

Syntax	Use Case	new Use
<code>int[] a = {1, 2, 3};</code>	ඉක්මන් initialize එකකට	Java internally uses new
<code>int[] a = new int[5];</code>	size එකක් දීලා පස්සේ values දාන්ත	new ලියන්නම ඕනෑම

● උදාහරණ:

```
int[] a = {10, 20, 30}; //direct initialization
int[] b = new int[3]; //array එකක් හදනවා, values පස්සේ දාන්න
b[0] = 10;
b[1] = 20;
b[2] = 30;
```

නිගමනය:

- **new keyword** එකක් Java එකට කියනවා "මට memory එකක් allocate කරන්න" කියලා
- Arrays හඳුදී, size එකක් නම් new අනිවාර්යයි
- Values දිලාම initialize කරනවා නම් new ලියන්නේ නැහැ **Java internally use** කරනවා

```
import java.util.*;
class Main {
    public static void main(String[] args) {

        byte[] b = new byte[5];
        System.out.println("byte: " + Arrays.toString(b));

        short[] s = new short[5];
        System.out.println("short: " + Arrays.toString(s));

        int[] i = new int[5];
        System.out.println("int: " + Arrays.toString(i));

        long[] l = new long[5];
        System.out.println("long: " + Arrays.toString(l));

        double[] d = new double[5];
        System.out.println("double: " + Arrays.toString(d));

        float[] f = new float[5];
        System.out.println("float: " + Arrays.toString(f));

        boolean[] bool = new boolean[5];
        System.out.println("boolean: " + Arrays.toString(bool));

        char[] c = new char[5];
        System.out.println("char: " + Arrays.toString(c));
```

```
String[] str = new String[5];
System.out.println("String: " + Arrays.toString(str));
}
}
byte: [0, 0, 0, 0, 0]
short: [0, 0, 0, 0, 0]
int: [0, 0, 0, 0, 0]
long: [0, 0, 0, 0, 0]
double: [0.0, 0.0, 0.0, 0.0, 0.0]
float: [0.0, 0.0, 0.0, 0.0, 0.0]
boolean: [false, false, false, false, false]
char: [ , , , , ] //mukuth na kna ekak dnne char wala efault alue widihat
String: [null, null, null, null, null]
```

- Methana wenne data tpe ala default values deela thiyyena eka

 ඔයාට array එක් values custom දාන්න ඕනෑම:

```
int[] x = new int[5];
x[0] = 10;
x[1] = 20;
System.out.println(Arrays.toString(x)); // [10, 20, 0, 0, 0]
```

◆ Array Creation Expression With initializer,

You create a new array using the **new** keyword, and at the same time, **you provide the initial values for the array using { }.**

```
int[] x = new int[]{1 , 2, 3, 4, 5};  
String[] names = new String[]{"Hi", "Hello", "WOW"};
```

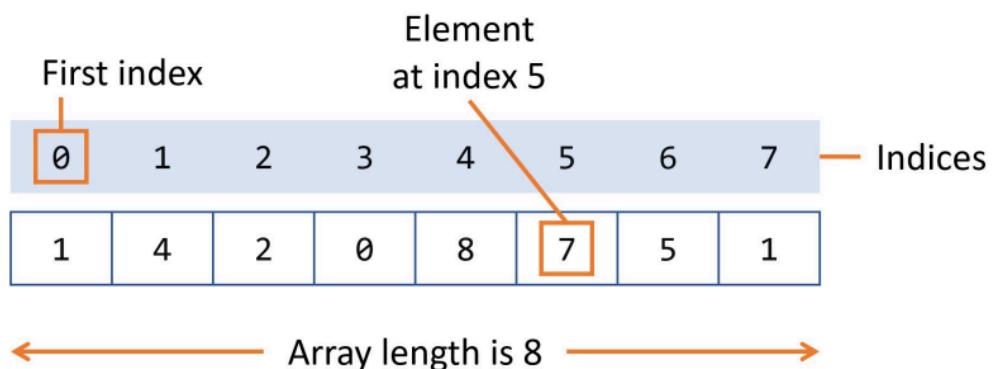
```
import java.util.*;  
class Main {  
    public static void main(String[] args) {  
        int[] x = new int[]{10, 20, 30, 40, 50};  
        System.out.println(x);  
        System.out.println(x.length);  
        System.out.println(Arrays.toString(x));  
    }  
}  
[I@6acbcfc0  
5  
[10, 20, 30, 40, 50]
```

```
import java.util.*;  
class Main {  
    public static void main(String[] args) {  
        int[] x = new int[]{10, 20, 30, 40, 50};  
        x = new int[] {88, 76, 54};  
        System.out.println(x.length);  
        System.out.println(Arrays.toString(x));  
    }  
}  
3  
[88, 76, 54]
```

Mehema danmama wenne kalin variable wala wagema x ta assign karala iwara wela nawatha assign karama x ta thawath array ekak use karala wenama value sign karala thibbama xge warthamana values wenne anthimata assign karapu value tika

```
class Main {  
    public static void main(String[] args) {  
        int x;  
        x = 10;  
/*  
int[] a;  
a = {10, 20, 30, 40, 50}; // illegal  
        int[] b;  
        b = new int[5];  
        int[] c;  
        c = new int[]{100, 200, 300};  
    }  
}
```

Java Array indexing



Java Array Item Reading(Accessing)

x[index]

```
int[] x = {10, 20, 30, 40, 50};  
S.O.P(x[0]); // 10  
S.O.P(x[1]); // 20  
S.O.P(x[4]); // 50
```

Java Arrays Do NOT Have Negative Indexing

```
int[] numbers = {10, 20, 30, 40, 50};  
System.out.println(numbers[-1]); // ✗ Error
```



Rina index agayan java wala na, namuth wenath language wala thinawa
Ex:python me widihatath thinawa rina widihatath thinawa

Python Array Indexing

Python Arrays Have Negative Indexing



```
numbers = [10, 20, 30, 40, 50]  
print(numbers[-1]) # ✅ Outputs: 50 (last element)
```

What is element and index ?

```
0 , 1 , 2 , 3 , 4  
int[] numbers = {10, 20, 30, 40, 50};  
----->  
          Array Length
```

මෙහි:

10 → 1වෙති element (index 0)

20 → 2වෙති element (index 1)

30 → 3වෙති element (index 2)

40 → 4වෙති element (index 3)

50 → 5වෙති element (index 4)

- **Last index = Array Length - 1**

Array Length = List index +1

numbers[0] ->10
numbers[1] ->20
numbers[2] ->30
numbers[3] ->40
numbers[4] ->50
numbers[5] -> error

numbers.length ->5

Numbers array eke awayawa ganana

Numbers[numbers.length-1] -> numbers[4] -> 50

Numbers array eke last index eka (anthima awayawaya)

● Print array value

Index eka use karala meka karanne

```
class Main {  
    public static void main(String[] args) {  
        int[] x = {100, 200, 300, 400, 500, 600, 700, 800};  
        System.out.println(x[0]); // 100  
        System.out.println(x[1]); // 200  
        System.out.println(x[2]); // 300  
    }  
}
```

100

200

300

◆ **ArrayIndexOutOfBoundsException**



- array range eke nathi index ekak print karanna awama me error eka enawa (me error ekata kina me nama mathaka thiyyaganna)

ArrayIndexOutOfBoundsException is a **runtime error** that happens when you try to access an array with an invalid index.

```
int[] x = new int[]{1, 2, 3, 4, 5};  
S.O.P(x[5]); // ❌ Error: ArrayIndexOutOfBoundsException
```

Me error eka compile karaddi awulak une na namuth error eka awe run karaddi nam meka

Ema nisa **Meka run time error ekak**

Runtime error - run karaddi ena error

Compile error - compile karaddi ena error

methana error ekk enne api methana element 8k witharai dala thiyyenne ekiyanne index 7i api index 8 illuwama javata hithaganna ba

```
import java.util.Arrays;  
class Main {  
    public static void main(String[] args) {  
        int[] x = {100, 200, 300, 400, 500, 600, 700, 800};  
        System.out.println(x[0]);  
        System.out.println(x[1]);  
        System.out.println(x[2]);  
        System.out.println(x[8]); // error  
        // methana error ekk enne api methana element 8k witharai dala thiyyenne ekiyanne index 7i api index 8 illuwama javata hithaganna ba  
    }  
}
```

```
100  
200  
300  
Exception // error
```

methana error ekk enne api methana element 8k witharai dala thiyanne ekiyanne
index 7i api index 8 illuwama javata hithaganna ba, ewagema -1 index ekakuth nathi
nsa javata hithaganna bari nisa error ekak denawa

```
class Main {  
    public static void main(String[] args) {  
        int[] x = {100, 200, 300, 400, 500, 600, 700, 800};  
        System.out.println(x[7]); // 800  
        System.out.println(x[0]); // 100  
        System.out.println(x[8]); // Error -  
java.lang.ArrayIndexOutOfBoundsException  
        System.out.println(x[-1]); // Error -  
java.lang.ArrayIndexOutOfBoundsException  
    }  
}
```

800

100

DAY 9

```
class Main {  
    public static void main(String[] args) {  
        int[] x = {100, 200, 300, 400, 500, 600, 700, 800};  
        int sum = x[0] + x[7];  
        System.out.println("Sum: " + sum);  
    }  
}
```

Sum: 900

```
class Main {  
    public static void main(String[] args) {  
        int[] x = {100, 200, 300, 400, 500, 600, 700, 800};  
        int sum = x[0] + x[x.length - 1]; //100+800 = 900  
        System.out.println("Sum: " + sum);  
    }  
}
```

Sum: 900

Use java Loops to Read Java Array Items

```
int[] x = {10, 20, 30, 40, 50};  
for (int i = 0; i < x.length; i++) {  
    System.out.println(x[i]);  
}
```

i=0
X[i] ->x[0]
i= 4
X[i] ->x[4]

Mehema danmama siyalu walu tika one by one print karagannawa for loop eka yatathe

```
import java.util.Arrays;
class Main {
    public static void main(String[] args) {
        int[] x = {10, 20, 30, 40, 50, 60, 70, 80};
        for (int i= 0; i<x.length ;i++) {
            System.out.println(x[i]);
        }
    }
}
10
20
30
40
50
60
70
80
```

```
import java.util.Arrays;
class Main {
    public static void main(String[] args) {
        int[] x = {10, 20, 30, 40, 50, 60, 70, 80, 90, 100,
110, 120, 200, 300};
        for (int i= 0; i<x.length ;i++) {
            System.out.print(x[i] + " ");
        }
    }
}
```

10 20 30 40 50 60 70 80 90 100 110 120 200 300

Apasata print karn widiha

```
import java.util.Arrays;
class Main {
    public static void main(String[] args) {
        int[] x = {3, 5, 6, 19, 2, 1, 11, 43, 55, 76, 80};
        for(int i = x.length-1; i>=0; i--) {
            System.out.println(x[i]);
        }
    }
}
```

```
80  
76  
55  
43  
11  
1  
2  
19  
6  
5  
3
```

Array eka string ekak widihata print karana part eka Arrays.toString() use karanne nathuwa gahanna

```
import java.util.Arrays;  
  
class Main {  
    public static void main(String[] args) {  
  
        int [] num = {34, 32 , 45 , 445};  
        System.out.print("[");  
  
        for(int i = 0; i < num.length; i++){  
            System.out.print(num[i] + ", ");  
        }  
  
        System.out.println("\b\b]");  
    }  
}
```

```
[34, 32, 45, 445]
```

```
class Main {  
    public static void main(String[] args) {  
        String[] str = {"Sri Lanka", "India", "UAE", "UK",  
"USA"};  
        /*
```

```

        str[0] -> Sri Lanka
        str[1] -> india
        str[2] -> UAE
        str[3] -> UK
        str[4] -> USA
    */
    for(int i = 0; i<str.length; i++) {
        System.out.println(str[i]);
    }
}

```

Sri Lanka
India
UAE
UK
USA

Odd value print

```

import java.util.Arrays;
class Main {
    public static void main(String[] args) {
        int[] x = {10, 11, 20, 21, 30, 31, 40, 41, 50, 51};
        for (int i= 0; i<x.length ; i++){
            if (x[i] % 2 != 0){
                System.out.println(x[i]);
            }
        }
    }
}

```

Me widiht hari

```

import java.util.Arrays;
class Main {
    public static void main(String[] args) {
        int[] x = {10, 11, 20, 21, 30, 31, 40, 41, 50, 51};
    }
}

```

```
        for (int i= 0; i<x.length ; i++) {
            if (x[i] % 2 == 1) {
                System.out.println(x[i]);
            }
        }
}
11
21
31
41
51
```

Q

```
class Main {
    public static void main(String[] args) {
        int[] x = {3, 5, 6, 19, 2, 1, 11, 43, 55, 76, 80};
        // write code to only print even numbers in the array
    }
}
```

```
class Main {
    public static void main(String[] args) {
        int[] x = {3, 5, 6, 19, 2, 1, 11, 43, 55, 76, 80};
        // write code to only print even numbers in the array
        for(int i = 0; i<x.length; i++) {
            if(x[i]%2==0) {
                System.out.println(x[i]);
            }
        }
    }
}
```

◆ For-each Loop in Java

The for-each loop (also called an enhanced for loop) is a special, **simple way to go through each item in an array or collection without worrying about indexes.**

```
int[] numbers = {10, 20, 30, 40};

for (int num : numbers) {
    System.out.println(num);
}
```

Meka arrays walata witharai use wenne

Meka loop wenne na

Array ekaka value one by one aran dena/Access karana wade karanawa

```
for (int item : x) {}
```

Methana x wisin araganna item one by one dagannawa me item kiyana variable

name ekata

Array type ekama thamai mthana variable type keats denneth

```
import java.util.Arrays;
class Main {
    public static void main(String[] args) {
        int[] x = {10, 11, 20, 21, 30, 31, 40, 41, 50, 51};
        for (int item : x){
            System.out.println(item);
        }
    }
}
10
11
20
21
30
31
40
41
50
51
```

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

```
        int[] x = {100, 200, 300, 400, 500};
        for(int num : x) {
            System.out.println(num);
        }
    }
}
100
200
300
400
500
```

Od value tika print karagamu for-each use karala

```
import java.util.Arrays;
class Main {
    public static void main(String[] args) {
        int[] x = {10, 11, 20, 21, 30, 31, 40, 41, 50, 51};
        for (int item : x){
            if (item % 2 != 0) {
                System.out.println(item);
            }
        }
    }
}
11
21
31
41
51
```

◆ Update Arrays items

You can update (change) an element in an array by:

Step 01: Accessing the element by its index

Step 02: Assigning a new value to that index

```
int[] numbers = {10, 20, 30, 40};
```

```
numbers[1] = 200;
```

```
import java.util.*;
class Main {
    public static void main(String[] args) {
        int[] x = {100, 200, 300, 400, 500};
        System.out.println(Arrays.toString(x));
        x[2] = 6000;
        System.out.println(Arrays.toString(x));
    }
}
[100, 200, 300, 400, 500]
[100, 200, 6000, 400, 500]
```

X[4] = x[4]+5

```
import java.util.Arrays;
class Main {
    public static void main(String[] args) {
        int[] x = {10, 20, 30, 40, 50};
        System.out.println(Arrays.toString(x));
        x[0] = 1000;
        x[1] = 2000;
        x[2] = 3000;
        x[3] = 4000;
        System.out.println(Arrays.toString(x));
        x[4] = x[4]+5;
        System.out.println(Arrays.toString(x));
    }
}
```

```
}
```

```
import java.util.Arrays;
class Main {
    public static void main(String[] args) {
        int[] x = {10, 20, 30, 40, 50};
        System.out.println(Arrays.toString(x));
        x[0] = 1000;
        x[1] = 2000;
        x[2] = 3000;
        x[3] = 4000;
        System.out.println(Arrays.toString(x));
        x[4] = x[4]+5;
        System.out.println(Arrays.toString(x));
        x[2] = x[2]+10;
        System.out.println(Arrays.toString(x));
    }
}
```

```
import java.util.Arrays;
class Main {
    public static void main(String[] args) {
        int[] x = {10, 20, 30, 40, 50};
        System.out.println(Arrays.toString(x));
        x[0] = 1000;
        x[1] = 2000;
        x[2] = 3000;
        x[3] = 4000;
        System.out.println(Arrays.toString(x));
        x[4] = x[4]+5;
        System.out.println(Arrays.toString(x));
        x[2] = x[3]+10;
        System.out.println(Arrays.toString(x));
    }
}
```

```
import java.util.*;
```

```

class Main {
    public static void main(String[] args) {
        int[] x = {100, 200, 300, 400, 500};
        System.out.println(Arrays.toString(x));
        for(int i = 0; i<x.length; i++) {
            x[i] = x[i]*2;
        }
        System.out.println(Arrays.toString(x));
    }
}

```

[100, 200, 300, 400, 500]
[200, 400, 600, 800, 1000]

◆ Array reverse (swap)

part1

```

import java.util.*;
class Main {
    public static void main(String[] args) {
        int[] x = {10, 20, 30, 40, 50};
        System.out.println(Arrays.toString(x)); // [10, 20, 30,
40, 50]
        // temp array
        int[] temp = new int[x.length]; // [0, 0, 0, 0, 0] / 5
    }
}

```

```

        for(int i = 0; i<temp.length; i++) {
            temp[i] = x[x.length - 1 - i];
            //temp 0 = x[5-1-0]-> x[4] <-x[x.length-1]
            //temp 1 = x[5-1-1]-> x[3] <-x[x.length-2]
            //temp 2 = x[5-1-2]-> x[2] <-x[x.length-3]
            //temp 3 = x[5-1-3]-> x[1] <-x[x.length-4]
            //temp 4 = x[5-1-4]-> x[0] <-x[x.length-5]
        }

        x = temp;

        System.out.println(Arrays.toString(x));
    }
}

```

HW

Part 2

Temp ona na ethakota

Array eken length eken hari bagayak loop wela thamai me wd wenne

```

import java.util.*;
class Main {
    public static void main(String[] args) {
        int[] x = {10, 20, 30, 40, 50};
        System.out.println(Arrays.toString(x)); // [10, 20, 30,
40, 50]

        // reverse the array (Only use x array)

        System.out.println(Arrays.toString(x));
    }
}

```

```

void reverseArray(int[] arr) {
    for (int i = 0; i < arr.length / 2; i++) {
        int temp = arr[i];
        arr[i] = arr[arr.length - 1 - i];
        arr[arr.length - 1 - i] = temp;
    }
}

```

```
import java.util.*;
```

```
class Main {  
    public static void main(String[] args) {  
        int[] x = new int[]{100, 200, 300, 400, 500};  
        int[] temp = new int[x.length]; // [0, 0, 0, 0]  
        int z = 0;  
        for(int i = x.length-1; i>=0; i--) {  
            System.out.println(x[i]);  
            temp[z] = x[i];  
            z++;  
        }  
        System.out.println(Arrays.toString(temp));  
        x = temp;  
        System.out.println(Arrays.toString(x));  
    }  
}
```

```
import java.util.*;  
  
class Main {  
    public static void main(String[] args) {  
        int[] x = new int[]{100, 200, 300, 400, 500};  
        int[] temp = new int[x.length]; // [0, 0, 0, 0]  
  
        for(int i = x.length-1; i>=0; i--) {  
            System.out.println(x[i]);  
            temp[(x.length-1) - i] = x[i];  
        }  
        System.out.println(Arrays.toString(temp));  
        x = temp;  
        System.out.println(Arrays.toString(x));  
    }  
}
```

Values swap

```
public class SwapExample {  
    public static void main(String[] args) {  
        int[] arr = {10, 20, 30, 40, 50};  
  
        // index 1 සහ index 3 තියන values swap කරමු  
        int temp = arr[1];  
        arr[1] = arr[3];  
        arr[3] = temp;  
  
        // result print කරමු  
        for(int num : arr) {  
            System.out.print(num + " ");  
        }  
    }  
}
```

◆ Length eka wadi karaganna vidiha

Array eke length eka fix, wenas karanna ba emarray ekama,

Array ekaka length eka wenas karaganna wenne aluth array ekaka value ekata dala anekuth value manually print karala

Ema array eka mul array eke namata samana karama apata array eke length eka wadi kara kiyala pennum kranna puluwan

```
import java.util.*;  
  
class Main {  
  
    public static void main(String[] args) {  
  
        String[] s = {"A", "B", "C"};  
  
        System.out.println(Arrays.toString(s));  
  
        String[] newArray = new String[6];// [null, null, null,  
null, null, null]
```

```
for(int i=0; i<s.length; i++) {  
    newArray[i] = s[i];  
}  
  
newArray[3] = "D";  
  
newArray[4] = "E";  
  
newArray[5] = "F";  
  
// System.out.println(Arrays.toString(newArray));  
  
s = newArray;  
  
System.out.println(Arrays.toString(s));  
}  
}
```

◆ Push and pop array

Push-array eke anthimata value ekak add karanawa

Pop kiyanne array ekk anthima value eka return karanawa

Mage eka,

```
import java.util.*;  
  
class Main {  
    public static void main(String[] args) {  
        String[] s = {"A", "B", "C"};  
        System.out.println(Arrays.toString(s));  
        s = push("D", s);  
        System.out.println(Arrays.toString(s));  
        s = pop(s);  
        System.out.println(Arrays.toString(s));  
    }  
  
    public static String[] push(String item, String[] s1) {  
        String[] s2 = new String[s1.length + 1];  
        for (int i = 0; i < s1.length; i++) {  
            s2[i] = s1[i];  
        }  
        s2[s1.length] = item;  
        return s2;  
    }  
  
    // api aluth array ekak hdnawe arry eke length ek ape  
    parana rray eke length ekt wada ekak duwen hadanawa  
    public static String[] pop(String[] s1) {
```

```

        if (s1.length == 0) {
            return s1;
        }
        String[] s2 = new String[s1.length - 1];// apata ekk
adu karanna ona nisa
        for (int i = 0; i < s2.length; i++) {
            s2[i] = s1[i];
        }
        return s2;
    }
}

```

Sirge eka

```

import java.util.*;
class Main {
    public static String[] push(String item, String[] arr) {
        String[] newArr = new String[arr.length + 1];
        for(int i = 0; i<arr.length; i++) {
            newArr[i] = arr[i];
        }
        newArr[newArr.length - 1] = item;
        return newArr;
    }

    public static String[] pop(String[] arr) { // length -> 3
        String[] newArr = new String[arr.length - 1]; // length
-> 2
        for(int i = 0; i<newArr.length; i++) {
            newArr[i] = arr[i];
        }
        return newArr;
    }

    public static void main(String[] args) {
        String[] s = {"A", "B", "C"};
        System.out.println(Arrays.toString(s));
        s = pop(s);
        System.out.println(Arrays.toString(s));
        s = pop(s);
        System.out.println(Arrays.toString(s));
        s = pop(s);
        System.out.println(Arrays.toString(s));
    }
}

```

```
s = push("D", s);
System.out.println(Arrays.toString(s));
s = push("E", s);
System.out.println(Arrays.toString(s));
s = push("F", s);
System.out.println(Arrays.toString(s));
}
}
```

Existing array assign to another variable

```
int[] x = {10, 20, 30, 40};
int[] y = x;
```

```
import java.util.*;
class Main {
    public static void main(String[] args) {
        String x = "Hello";
        String y = x;
    }
}
```

```
        System.out.println(x);
        System.out.println(y);
        y = "hiiii";
        System.out.println(x);
        System.out.println(y);
    }
}
```

Hello
Hello
Hello
hiiii

```
import java.util.*;
class Main {
    public static void main(String[] args) {
        String[] s = {"A", "B", "C", "D", "E", "F"};
        String[] newS = s;
        System.out.println("s - " + Arrays.toString(s));
        System.out.println("new s - " + Arrays.toString(newS));

        newS[0] = "Hello";
        System.out.println("s - " + Arrays.toString(s));
        System.out.println("new s - " + Arrays.toString(newS));

        s[s.length-1] = "Hiiii";
        System.out.println("s - " + Arrays.toString(s));
        System.out.println("new s - " + Arrays.toString(newS));
    }
}
```

s - [A, B, C, D, E, F]
new s - [A, B, C, D, E, F]
s - [Hello, B, C, D, E, F]
new s - [Hello, B, C, D, E, F]
s - [Hello, B, C, D, E, Hiiii]
new s - [Hello, B, C, D, E, Hiiii]

```
import java.util.*;
class Main {
    public static void main(String[] args) {
```

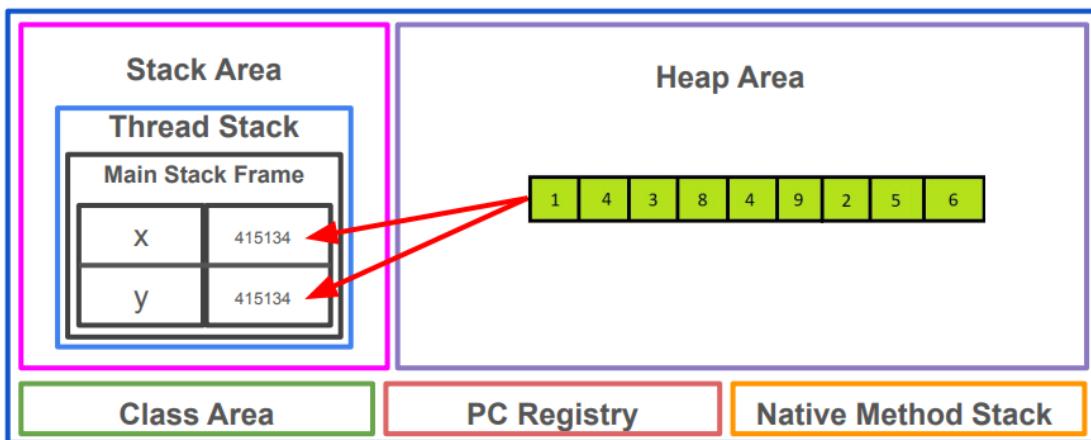
```

/*
int a = 10;
int b = a;
System.out.println("1st Round - a: " + a);
System.out.println("1st Round - b: " + b);
b = 20;
System.out.println("2st Round - a: " + a);
System.out.println("2st Round - b: " + b);
*/
    int[] x = new int[]{100, 200, 300, 400, 500};
    int[] y = x;
    System.out.println("1st Round - x: " +
Arrays.toString(x));
    System.out.println("1st Round - y: " +
Arrays.toString(y));
    y[0] = 8000;
    System.out.println("2nd Round - x: " +
Arrays.toString(x));
    System.out.println("2nd Round - y: " +
Arrays.toString(y));
    y[3] = 9000;
    System.out.println("3rd Round - x: " +
Arrays.toString(x));
    System.out.println("3rd Round - y: " +
Arrays.toString(y));
    x[4] = 0;
    System.out.println("4th Round - x: " +
Arrays.toString(x));
    System.out.println("4th Round - y: " +
Arrays.toString(y));
}
}

```

1st Round - x: [100, 200, 300, 400, 500]
1st Round - y: [100, 200, 300, 400, 500]
2nd Round - x: [8000, 200, 300, 400, 500]
2nd Round - y: [8000, 200, 300, 400, 500]
3rd Round - x: [8000, 200, 300, 9000, 500]
3rd Round - y: [8000, 200, 300, 9000, 500]
4th Round - x: [8000, 200, 300, 9000, 0]
4th Round - y: [8000, 200, 300, 9000, 0]

Java Array in Java Memory



Mehema ekama memory eka refer karala thina nisa thamai x wenas karama yth wenas wenne

What is the solution?

1. Use **Arrays.copyOf(array, array_length)**
2. Use **.clone()** method

```
int[] x = {10, 20, 30, 40};  
  
int[] y = Arrays.copyOf(x, x.length);  
  
int[] z = x.clone();
```

Apata mekata karanna wena solution eka wenne aluthen array ekak wenama hadala ema values ema array eka thulata insert karana eka

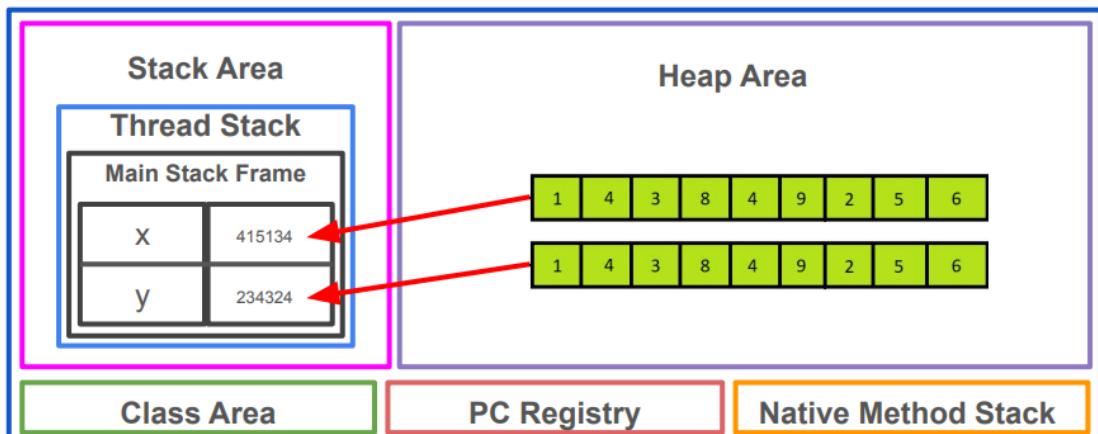
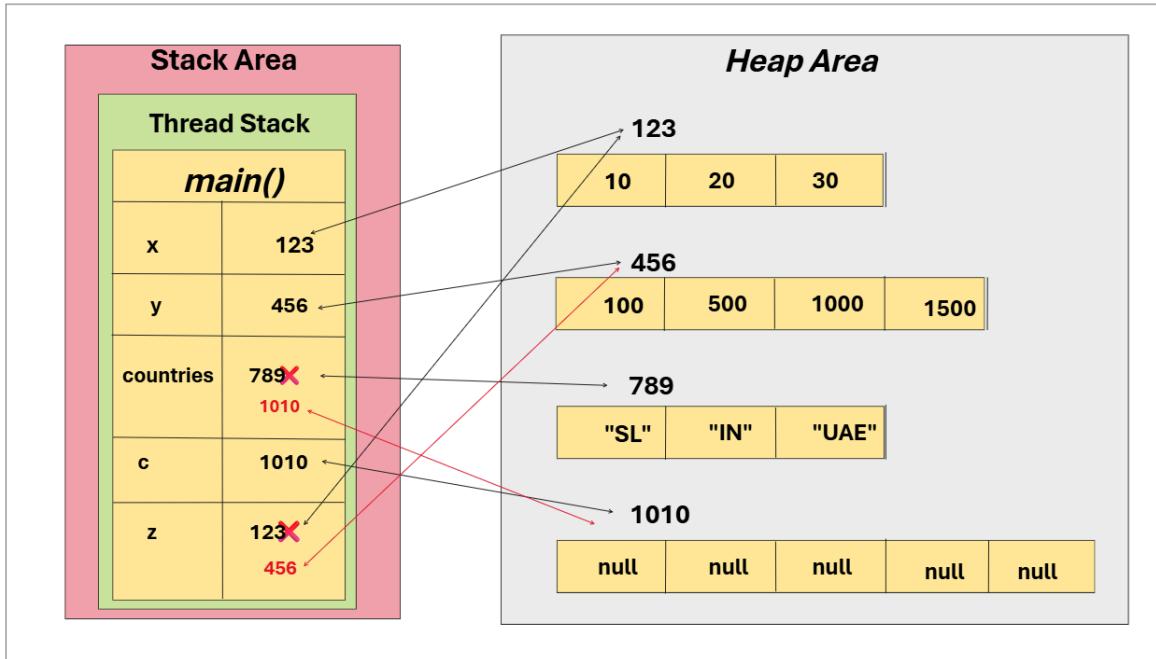
◆ Make an array copy

Arrays wala length eke wenasak karanna wenawanam api me wada karanawa

- 1) **Arrays.copyOf(array, array_length)**

2) clone

Java Memory



Java Array in Memory

1) Arrays.copyOf(array, array_length)

```
import java.util.*;
class Main {
    public static void main(String[] args) {
        String[] s = {"A", "B", "C", "D", "E", "F"};
        // String[] newS = Arrays.copyOf(s, s.length);
        String[] newS = s.clone();
        System.out.println("s - " + Arrays.toString(s));
```

```
        System.out.println("new s - " + Arrays.toString(newS));

        newS[0] = "Lanka";
        System.out.println("s - " + Arrays.toString(s));
        System.out.println("new s - " + Arrays.toString(newS));
    }
}
```

```
import java.util.*;

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

        int[] x = new int[]{100, 200, 300, 400, 500};
        int[] y = Arrays.copyOf(x, x.length);
        System.out.println("1st Round - x: " +
Arrays.toString(x));
        System.out.println("1st Round - y: " +
Arrays.toString(y));
    }
}
```

```
import java.util.*;

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

        int[] x = new int[]{100, 200, 300, 400, 500};
        int[] y = Arrays.copyOf(x, x.length);
```

```

        System.out.println("1st Round - x: " +
Arrays.toString(x));
        System.out.println("1st Round - y: " +
Arrays.toString(y));
        y[0] = 8000;
        System.out.println("2nd Round - x: " +
Arrays.toString(x));
        System.out.println("2nd Round - y: " +
Arrays.toString(y));

        x[4] = 6000;
        System.out.println("3rd Round - x: " +
Arrays.toString(x));
        System.out.println("3rd Round - y: " +
Arrays.toString(y));

    }
}

```

```

import java.util.*;

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

        int[] x = new int[]{100, 200, 300, 400, 500};
        int[] y = x.clone();
        System.out.println("1st Round - x: " +
Arrays.toString(x));
        System.out.println("1st Round - y: " +
Arrays.toString(y));
        y[0] = 8000;
        System.out.println("2nd Round - x: " +
Arrays.toString(x));
        System.out.println("2nd Round - y: " +
Arrays.toString(y));

        x[4] = 6000;
    }
}

```

```

        System.out.println("3rd Round - x: " +
Arrays.toString(x));
        System.out.println("3rd Round - y: " +
Arrays.toString(y));
    }
}

```

2)clone

Int[] x = {10, 20, 30}

Int[] y x.clone

```
///////////////////////////////
//
```

HW

```
///////////////////////////////
//
```

```

import java.util.*;

class Main {

    public static void main(String[] args) {

        // Ex 01

        int[] x = {10, 40, 50, 5, 70, 41, 3, 8};

        // 1. Largest Number ? (Use a Loop)

        // 2. Smallest Number ? (Use a Loop)

        // 3. Find the sum of largest and the smallest numbers

        in the array
    }
}

```

// Ex 02

```
int[] y = {100, 200, 20, 31, 100, 21, 20, 300, 20};  
// 1. find duplicates?  
  
// Ex 03  
  
int[] a = {10, 20, 100, 300, 400, 500, 550, 617};  
int[] b = {10, 20, 100, 300, 411, 500, 550, 617};  
int[] c = {10, 20, 100, 300, 400, 500, 550, 617};  
// 1. a equal b ?  
// 2. a equal c ?  
// 3. b equal c ?  
  
// Ex 04  
  
int[] j = {10, 40, 50, 5, 70, 41, 3, 8};  
// 1. sort this array (smallest -> largest) |  
System.out.println(Arrays.toString(j));  
// 2. sort this array (largest -> smallest) |  
System.out.println(Arrays.toString(j));  
}  
}
```

Can You Change the Length of a Java Array?



No, once you create an array in Java, its length is fixed.

But is there a solution?

Yes! Let's find out it.



```
int[] oldArray = {1, 2, 3};  
int[] newArray = new int[5];  
  
// Copy old values to new array  
for (int i = 0; i < oldArray.length; i++)  
{  
    newArray[i] = oldArray[i];  
}  
  
S.O.P(Arrays.toString(newArray));
```

◆ Multidimensional Array

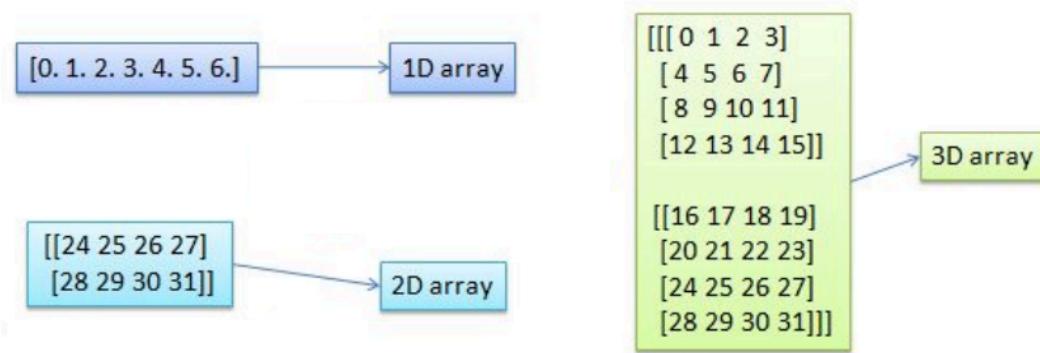
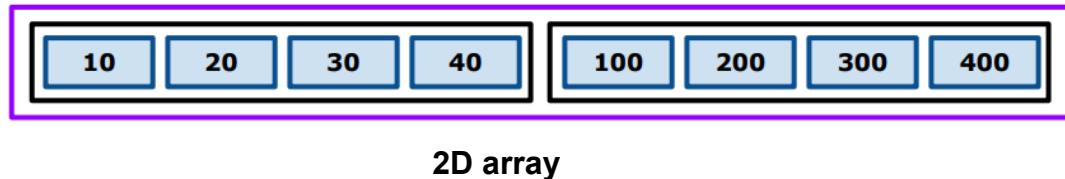
Arrays kihipayak ekathu karala group karanawa

[] 1D array

[][] 2D array

[][][] 3D array

A multidimensional array is essentially an array of arrays.



● Arrays.deepToString()

Multi dimensional print karaganne me widihata

```
import java.util.*;
class Main {
    public static void main(String[] args) {

        int[] x = {10, 40, 50, 5, 70, 41, 3, 8}; // 1d , 2d ,
3d??? // 1D

        int[][] y = {{10, 20, 30, 40}, {100, 200, 300, 400},
{1000, 2000, 3000, 4000}}; // 2D
        System.out.println(y.length); // 3
        System.out.println(Arrays.deepToString(y));
    }
}
```

```

import java.util.*;
class Main {
    public static void main(String[] args) {
        int[] x = {10, 20, 30, 40}; //1D array
        int[][] y = {{10, 20, 30}, {40, 50, 60}, {70, 80, 90}};
//2D array
        System.out.println(y); // memory address e.g.
[[I@2a139a55
        System.out.println(Arrays.toString(y)); // array with
memory addresses e.g. [[I@15db9742, [I@6d06d69c, [I@7852e922]
        System.out.println(Arrays.deepToString(y)); // [[10,
20, 30], [40, 50, 60], [70, 80, 90]]
    }
}

```

```

class Main {
    public static void main(String[] args) {
        int[][] y = {{10, 20, 30}, {40, 50, 60}, {70, 80, 90}};
        System.out.println(y[0]); // memory address memory
address eka enne api methanadi print karanna adahaskaranne
y[0] array eka nisa
        System.out.println(y[0][0]); // 10
        System.out.println(y[0][1]); // 20
        System.out.println(y[0][2]); // 30
        System.out.println(y[1][0]); // 40
        System.out.println(y[1][1]); // 50
        System.out.println(y[1][2]); // 60
        System.out.println(y[2][0]); // 70
        System.out.println(y[2][1]); // 80
        System.out.println(y[2][2]); // 90
    }
}

```

y[0][0]
0 index kenage 0 index value eka denna

Methanadi Arrays.toString use karanne natthe api methanadi ganne array eka
athule thina value eka nisa

```
import java.util.Arrays;

class Main {
    public static void main(String[] args) {
        String [][] y = {
            {"Sri lanka", "india", "pakistan"},,
            {"UK", "France", "Italy"},,
            {"USA", "Maxiko", "Canada"},,
            {"africa", "madagasker"}}
        ;
        System.out.println(y.length);
        System.out.println(Arrays.deepToString(y)); //array ekak
athule multi dimention array ekakma print karanawa nam
Array.deepToString() use karanawa
        System.out.println(y[0]); // memory addres print //
print karanna hadanne array ekak nia Arrays.toString ona
karanawa
        System.out.println(Arrays.toString(y[0])); // [Sri lanka,
india, pakistan]
        System.out.println(Arrays.toString(y[1])); // [UK,
France, Italy]
        System.out.println(Arrays.toString(y[2])); // [USA,
Maxiko, Canada]
        System.out.println(Arrays.toString(y[3])); // [africa,
madagasker] // methanadi array ekak print karana nisa api
toString danawa
        System.out.println(y[0][0]); // api methanadi print
karganne value ekak nisa Arrays.toString use karanna ona na
        System.out.println(y[0][1]);
        System.out.println(y[0][2]);
        System.out.println(y[1][0]);
        System.out.println(y[1][1]);
        System.out.println(y[1][2]);
        System.out.println(y[2][0]);
        System.out.println(y[2][1]);
        System.out.println(y[2][2]);
        System.out.println(y[3][0]);
        System.out.println(y[3][1]);
    }
}
```

●2D array Value Print use for loop

Wena wenama print karanne nathuwa loop dala print karanawa nam

```
import java.util.Arrays;

class Main {
    public static void main(String[] args) {
        String [][] y = {
            {"Sri lanka", "india", "pakistan"}, 
            {"UK", "France", "Italy"}, 
            {"USA", "Maxiko", "Canada"}, 
            {"africa", "madagasker"} 
        };

        for (int i = 0; i < y.length; i++) {
            for (int j = 0; j < y[i].length; j++) {
                System.out.println(y[i][j]);
            }
        }
    }
}
```

```
import java.util.Arrays;

class Main {
    public static void main(String[] args) {
        String [][] y = {
            {"Sri lanka", "india", "pakistan"}, 
            {"UK", "France", "Italy"}, 
            {"USA", "Maxiko", "Canada"}, 
            {"africa", "madagasker"} 
        };

        System.out.println("-----");
        System.out.println("+ Countries +");

        for (int i = 0; i < y.length; i++) {
            for (int j = 0; j < y[i].length; j++) {
```

```
System.out.println("-----+");  
  
        System.out.print("+ "+y[i][j] );  
        System.out.println("");  
    }  
}  
  
}  
}
```

Rows `banana`,
`x.length`

Mul row eke thiyyena columns ganna
x[0].length

1st row eka print karanna

```
import java.util.Arrays;

class Main {
    public static void main(String[] args) {
        int [][] x = {{1,2,3,4,5},{2,6,7,4,5}};
        System.out.println(Arrays.toString(x[0]));
    }
}
```

Row columns ganan ganne index test ekata

Row 2 (look like index)= row element 3

Columns 3 (look like index) = column element 4

Habai api hadaddi rows 2i columns 2i kiwama row element 3i

column element 3i

2Row, 2column,

{

0 1 2

$$\{4, 3, 5\}, \quad 0$$

```

{6, 7, 9}, 1
{5, 3, 2} 2
};

```

Ewa use karaddi thamai indexing widihata use wenne

	Column 0	Column 1	Column 2
Row 0	x[0][0]	x[0][1]	x[0][2]
Row 1	x[1][0]	x[1][1]	x[1][2]
Row 2	x[2][0]	x[2][1]	x[2][2]

```
int[][] x = { {10, 20, 30}, {40, 50, 60}, {70, 80, 90} };
```

Row 0

```

x[0][0] -> 10
x[0][1] -> 20
x[0][2] -> 30

```

Row 1

```

x[1][0] -> 40
x[1][1] -> 50
x[1][2] -> 60

```

Row 2

```

x[2][0] -> 70
x[2][1] -> 80
x[2][2] -> 90

```

● Update Multidimensional Array

```

import java.util.Arrays;

class Main {
    public static void main(String[] args) {
        String [][] y = {

```

```

        {"Sri lanka", "india", "pakistan"},  

        {"UK", "France", "Italy"},  

        {"USA", "Maxiko", "Canada"},  

        {"africa", "madagasker"}  

    };  
  

    System.out.println(Arrays.deepToString(y));  
  

    y[1][2] = "Scotland";  

    System.out.println(Arrays.deepToString(y));  
  

}  

}

```

MAX

```

class Main {  

    public static void main(String[] args) {  
  

        int [] num = {34,32 ,45 ,445};  
  

        int max = num[0];  
  

        for (int i = 0; i < num.length; i++) {  

            if (num[i] > max) {  

                max = num[i];  

            }
        }
        System.out.println("Max : " + max);
    }
}

```

MAX & MIN & MAX + MIN

```

import java.util.*;  
  

public class Main{  
  

    public static void main(String[] args) {  


```

```

        int[] num = {20, 10, 40, 60, 50};

        /*
         1. Max ?
         2. Min ?
         3. Max + Min = ?

        */

        int max = num[0];

        int min = num[0];

        for(int i=0; i<num.length; i++) {

            if(num[i]>max) {

                max = num[i];
            }

            if(num[i]<min) {

                min = num[i];
            }
        }

        System.out.println("Max: " + max);

        System.out.println("Min: " + min);

        System.out.println("Total: " + (max + min));
    }
}

```

//////////////////////////////

2D Array -> MAX & MIN & MAX + MIN

```

import java.util.*;

public class Main{

```

```

public static void main(String[] args) {

    int[][] num = {{90, 30, 10}, {100, 5, 10}, {6, 12, 1},
{45, 5, 300}};

    for(int i=0; i<num.length; i++) {

        int max = num[i][0];
        int min = num[i][0];
        for(int j=0; j<num[i].length; j++) {
            if(num[i][j]>max) {
                max = num[i][j];
            }
            if(num[i][j]<min) {
                min=num[i][j];
            }
        }

        System.out.println("Max: " + max);
        System.out.println("Min: " + min);
        System.out.println("Total: " + (max + min));
        System.out.println("/////////////////////");
    }
}

```

/////////////////////////////

Duplicate Value

```
public class Main{
```

```
public static void main(String[] args) {  
  
    int[] x = {100, 200, 300, 100, 400, 90, 90, 50, 600, 50,  
300};  
  
    for(int i = 0; i<x.length - 1; i++) {  
  
        for(int j = i+1; j<x.length; j++) {  
  
            if(x[i] == x[j]) {  
  
                System.out.println("Duplicated: " + x[i]);  
  
            }  
  
        }  
  
    }  
  
}
```

//////////

```
public class Main{  
  
    public static boolean isFind(int[] arr, int value) {  
  
        for(int i=0; i<arr.length; i++) {  
  
            if(arr[i] == value) {  
  
                return true;  
  
            }  
  
        }  
  
        return false;  
  
    }  
  
    public static void main(String[] args) {
```

```
        int[] x = {100, 200, 300, 100, 400, 90, 90, 50, 600, 50,
300};

        boolean result = isFind(x, 90);

        System.out.println("Result: " + result);

    }

}
```

```
///////////////////////////////
//



import java.util.Scanner;

public class Main{

    public static boolean isFind(int[] arr, int value) {

        for(int i=0; i<arr.length; i++) {

            if(arr[i] == value) {

                return true;
            }
        }

        return false;
    }

    public static void main(String[] args) {

        int[] x = new int[5];

        Scanner input = new Scanner(System.in);

        for(int i=0; i<x.length; i++) {

            System.out.print("Enter element: ");

```

```
        int element = input.nextInt();

        x[i] = element;

    }

System.out.print("Enter finding value: ");

int findingValue = input.nextInt();

boolean result = isFind(x, findingValue);

System.out.println("Result: " + result);

}

}
```

Total 2d arrays

```
class Main{
    public static void main(String args[]){
        int ar[][]={{1,2},{1,2,3,4},{1,2,3,4,5},{1,2,3,4,5,6}};

        int total=0;
        for(int a[] : ar){
            for(int b : a){
                total += b;
            }
        }
        System.out.println("Total :" +total);
    }
}
```

```
class Main{
    public static void main(String args[]){
        int ar[][]={{1,2},{1,2,3,4},{1,2,3,4,5},{1,2,3,4,5,6}};

        int total=0;
        for(int a[] : ar){
            for(int i = 0; i < a.length; i++){
                total += a[i];
            }
        }
        System.out.println("Total :" +total);
    }
}
```

Day 4 remedial part (methana awashya note tikak thinawa ewa wen karala gnna)

```
import java.util.Arrays;

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

        int [] marks = {87, 90, 67, 50};
        myMethod("John", 30, marks);
```

```
    }

    public static void myMethod(String name, int age , int
[]marks) {
        System.out.println("Name : " + name);
        System.out.println("Age : " + age);
        System.out.println("Marks : " +
Arrays.toString(marks));
    }

}
```

```
import java.util.*;

class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        int value = 0;

        System.out.print("Enter the length of the array :");
        int length = input.nextInt();

        for (int i = 0; i < length; i++) {
            System.out.print("Enter Value :");
            value = input.nextInt();

            myMethod(length,value);
        }
    }

    public static void myMethod( int length , int value){
        int[] array = new int[length];
        for (int i = 0; i < array.length; i++) {
```

```
        array[i] = value;
    }

    System.out.println(Arrays.toString(array));
}

}
```

```
import java.util.*;
class Main{
    public static void main(String[] args){
        int[] s = new int[5];

        for(int i = 0;i<s.length;i++){
            Scanner input = new Scanner(System.in);
            System.out.print("input number :");
            int newnum = input.nextInt();
            s[i]=newnum;
        }

        System.out.println(Arrays.toString(s));
    }
}
```

```
import java.util.Arrays;
import java.util.Scanner;

public class Example {
    public static void myMethod() {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter length : ");
        int length = input.nextInt();
        int[] array = new int[length];

        for (int i = 0; i < array.length; i++) {
```

```
        System.out.println("Input value for index " + i + " : ");
        int value = input.nextInt();
        array[i] = value;
    }
    System.out.println(Arrays.toString(array));
}

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

```
import java.util.Arrays;
import java.util.Scanner;

class Example {
    public static void myMethod() {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter length : ");
        int length = input.nextInt();
        int[] marks = new int[length];

        for (int i = 0; i < marks.length; i++) {
            System.out.print("Input value for index " + i + " : ");
            int value = input.nextInt();
            marks[i] = value;
        }

        int total = 0;

        for (int i = 0; i < marks.length; i++) {
            System.out.println("marks[" + i + "] = " +
marks[i]);
            total = total + marks[i];
        }

        System.out.println(Arrays.toString(marks));
        System.out.println("Total = " + total);
    }
}
```

```
        System.out.println("Average = " + (double) total /  
marks.length);  
  
    }  
    public static void main(String[] args) {  
        myMethod();  
    }  
}
```

```
import java.util.Arrays;  
import java.util.Scanner;  
  
class Example {  
    public static void myMethod() {  
        Scanner input = new Scanner(System.in);  
  
        System.out.print("Enter length : ");  
        int length = input.nextInt();  
  
        int[] marks = new int[length];  
  
        for (int i = 0; i < marks.length; i++) {  
            System.out.print("Input value for index " + i + " :  
");  
            int value = input.nextInt();  
  
            marks[i] = value;  
        }  
    }  
}
```

```
}

int total = 0;

for (int i = 0; i < marks.length; i++) {
    System.out.println("marks[" + i + "] = " +
marks[i]);
    total = total + marks[i];
}

int max = marks[0];
for (int i = 1; i < marks.length; i++) {
    if (marks[i] > max) {
        max = marks[i];
    }
}

int min = marks[0];
for (int i = 1; i < marks.length; i++) {
    if (marks[i] < min) {
        min = marks[i];
    }
}

System.out.println(Arrays.toString(marks));
System.out.println("Total = " + total);
System.out.println("Average = " + (double) total /
marks.length);
System.out.println("Max Value = " + max );
System.out.println("Minimum Value = " + min);

}

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

```
enter two arry, and input one array insert these 2 arrays
```

Wrong

```
import java.util.Arrays;

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

        System.out.println("enter two arry, and input one array
insert these 2 arrays");
        int[] arr1 = {43, 45, 56, 67, 78};
        int[] arr2 = {34, 56, 67, 76, 66};

        int [] newArr = new int[arr1.length];

        for(int i=0;i<arr1.length;i++) {

            newArr[i] = arr1[i];
            newArr[i+1] = arr2[i];

        }
        System.out.println(Arrays.toString(newArr));
    }
}
```

Right

```
import java.util.*;
class Main{
    public static void main(String[]args) {
        int[] array1 = {10,20,30};
        int[] array2 = {40,50,60};

        int[] newArray =new int[array1.length + array2.length];

        for(int i = 0;i < array1.length;i++){
            newArray[i]=array1[i];
        }
```

```
        for(int i = 0;i<array2.length;i++) {
            newArray[array1.length + i]= array2[i];
        }
        System.out.println(Arrays.toString(newArray));
    }
}
```

```
import java.util.*;
class Main{
    public static void main(String[]args){
        int[] array1 = {10,20,30};
        int[] array2 = {40,50,60};

        int[] newArray =new int[array1.length + array2.length];

        for(int i = 0;i < array1.length;i++) {
            newArray[i]=array1[i];
        }
        for(int i = array1.length, j=0
; i<newArray.length;i++,j++) {
            newArray[ i]= array2[j];
        }
        System.out.println(Arrays.toString(newArray));
    }
}
```

Revers

Wrong

```
import java.util.*;
class Main{
    public static void revers() {

        String[] names = {"Nimal","Amal","Wimal","Saman"};

        for(int i = 0;i < names.length;i++) {
            String a = names[i];
```

```

        names[i] = names[names.length-1-i];
        names[names.length-1-i]=a;
    }
    System.out.println(Arrays.toString(names));
}

public static void main(String[] args) {
    revers();
}
}

```

[Saman, Wimal, Amal, Nimal]

Right

Aluth rray ekak hadannenathuw hadaana widiha

```

import java.util.*;
class Main{
    public static void revers() {

        String[] names = {"Nimal", "Amal", "Wimal", "Saman"};

        for(int i = 0;i < names.length/2;i++) {
            String a = names[i];
            names[i] = names[names.length-1-i];
            names[names.length-1-i]=a;
        }
        System.out.println(Arrays.toString(names));
    }

    public static void main(String[] args) {
        revers();
    }
}

```

[Saman, Wimal, Amal, Nimal]

04) Reverse given array.

Aluth array ekak aragena dana widiha

import java.util.Arrays;

```

public class Example {
    public static void reverse(String[] name) {
        String[] reverseNames = new String[name.length];
        for (int i = 0, j = name.length - 1; i < reverseNames.length; i++, j--) {

```

```

        reverseNames[i] = name[j];
    }

    System.out.println(Arrays.toString(reverseNames));
}
public static void main(String[] args) {
    String[] names = { "Nimal", "Amal", "Wimal", "Saman" };
    reverse(names);
}
}

```

Duplicate Check

Check duplicate value

```

import java.util.Arrays;

class Main {

    public static void main(String args[]) {
        checkDuplicate();
    }
    public static void checkDuplicate() {
        int [] y = { 100 , 200 , 20 , 31 , 100 , 21 , 20 , 300
, 20 };

        for ( int i = 0 ; i <= y. length ; i++) {
            for ( int j = i + 1 ; j < y. length ; j++) {
                if (y[i] != y[j]) {
                    continue;
                }
                System.out.println(y[i]);
                System.out.println(" index "+ i);
                System.out.println(" index " + j);

            }
        }
    }
}

```

```
import java.util.*;
```

```
class Main{
```

```
public static void checkDuplicates(int[] myarray){  
    for (int i = 0; i < myarray.length; i++) {  
        int j;  
        for (j = 0; j < i; j++) {  
            if (myarray[i] == myarray[j]) {  
                break;  
            }  
        }  
  
        if (j == i) {  
            for (int k = i + 1; k < myarray.length; k++) {  
                if (myarray[i] == myarray[k]) {  
                    System.out.println(myarray[i] + " is duplicate.");  
                    break;  
                }  
            }  
        }  
    }  
}
```

```
public static void main(String[] args){  
  
    int myArray[] = {100, 200, 20, 31, 100, 21, 20, 300, 20};  
    checkDuplicates(myArray);
```

```
}
```

```
import java.util.*;  
class Main {  
    public static void duplicate(){  
        int[] y = {5,8,5,3,5};  
  
        for(int i= 0; i <= y.length -1;i++){  
            int j;  
            for (j = 0; j < i; j++) {  
                if (y[i] == y[j]) {  
                    break;  
                }  
            }  
            if(j==i){for(int k = i+1;k <y.length;k++){  
                if(y[i] == y[k]){
```

```

        System.out.println("duplicate : "+y[i]);
        break;
    }
}
// System.out.println("index : "+i);
// System.out.println("index : "+k);

}
}
}

import java.util.Arrays;

class Main {

    public static void main(String args[]) {
        checkDuplicate();
    }

    public static void checkDuplicate() {
        int [] y = { 100 , 200 , 20 , 31 , 100 , 21 , 20 , 300
, 20 };

        for(int x: y){
            System.out.println(x);
        }
    }
}

```

2d array

```

class Main {
    public static void main(String[] args) {
        int[][] y = new int[3][3];

        y[0][0]=1;
        y[0][1]=2;
        y[0][2]=3;
        y[1][0]=4;
        y[1][1]=5;
        y[1][2]=6;
        y[2][0]=7;
        y[2][1]=8;
        y[2][2]=9;
    }
}

```

```
        for(int i=0;i<3;i++) {
            for(int j=0;j<3;j++) {
                System.out.print(y[i][j]+ " ");
            }
            System.out.println();
        }
    }
}
```

```
import java.util.*;

class Main{
    public static void checkDuplicates(int[] myarray) {
        for (int i = 0; i < myarray.length; i++) {
            int j;
            for (j = 0; j < i; j++) {
                if (myarray[i] == myarray[j]) {
                    break;
                }
            }

            if (j == i) {
                for (int k = i + 1; k < myarray.length; k++) {
                    if (myarray[i] == myarray[k]) {
                        System.out.println(myarray[i] + " is
duplicate.");
                    }
                }
            }
        }
    }

    public static void main(String[] args) {

        int myArray[] = {100, 200, 20, 31, 100, 21, 20, 300,
20};
        checkDuplicates(myArray);
    }
}
```

```
    }
}
```

```
public class Main {
    public static void main(String[] args) {
        int[][] a = {
            {1, 3},
            {5, 7},
            {9, 11},
            {13, 15}
        };

        // a. Print the second element of each row
        System.out.println("Second element of each row:");
        for (int i = 0; i < a.length; i++) {
            System.out.println(a[i][1]);
        }

        // b. Print All number numbers
        System.out.println("\nPrint All number numbers:");
        for (int i = 0; i < a.length; i++) {
            for (int j = 0; j < a[i].length; j++) {
                if (a[i][j] % 2 != 0) {
                    System.out.println(a[i][j]);
                }
            }
        }

        // c. Print only even numbers in the array
        System.out.println("\nEven numbers in the array:");
        for (int i = 0; i < a.length; i++) {
            for (int j = 0; j < a[i].length; j++) {

                if (a[i][j] % 2 == 0) {
                    System.out.println(a[i][j]);
                }
            }
        }
    }
}
```

```

// d. Print only odd numbers
System.out.println("\nOdd numbers in the array:");
for (int i = 0; i < a.length; i++) {
    for (int j = 0; j < a[i].length; j++) {
        if (a[i][j] % 2 != 0) {
            System.out.println(a[i][j]);
        }
    }
}

// e. Print all elements using a for-each loop
System.out.println("\nAll elements using for-each loop:");
for (int row[] : a){
    for (int num : row){
        System.out.println(num);
    }
}

// f. Print all elements in reverse row order
System.out.println("\nPrint all elements in reverse row order:");
for (int i = a.length - 1 ; i >= 0; i--){
    for (int j = 0; j < a[i].length; j++){
        System.out.println(a[i][j]);
    }
}

// g. Print all elements in reverse
System.out.println("\nPrint all elements in reverse:");
for (int i = a.length - 1 ; i >= 0; i--){
    for (int j = a[i].length-1; j >=0; j--){
        System.out.println(a[i][j]);
    }
}

// h. Print the number of rows (length of the array)
System.out.println("\nNumber of rows: " + a.length);

// i. Print the number of columns in each row
for (int i = 0; i < a.length; i++) {
    System.out.println("\nNumber of columns in row " + i + ": " +
a[i].length);
}

```

```

// j. Print the first row
System.out.print("\nFirst row: ");
for (int i = 0; i < a[0].length; i++) {
    System.out.print(a[0][i] + " ");
}

// k. Print the last row
System.out.print("\nLast row: ");
for (int i = 0; i < a[a.length - 1].length; i++) {

    System.out.print(a[a.length - 1][i] + " ");
}

// l. Print number of rows and columns
System.out.println("\nNumber of rows: " + a.length);
System.out.println("Number of columns: " + a[0].length); // assuming
all rows have the same length

// m. Print all values in the second row
System.out.print("\nSecond row: ");
for (int i = 0; i < a[1].length; i++) {
    System.out.print(a[1][i] + " ");
}

// n. Find and print the largest value in the array
int max = a[0][0]; // start with the first element
for (int i = 0; i < a.length; i++) {
    for (int j = 0; j < a[i].length; j++) {
        if (a[i][j] > max) {
            max = a[i][j];
        }
    }
}
System.out.println("\nLargest value in the array: " + max);

}

}

//Second element of each row:
3
7
11
15

```

Print All number numbers:

```
1  
3  
5  
7  
9  
11  
13  
15
```

Even numbers in the array:

Odd numbers in the array:

```
1  
3  
5  
7  
9  
11  
13  
15
```

All elements using for-each loop:

```
1  
3  
5  
7  
9  
11  
13  
15
```

Print all elements in reverse row order:

```
13  
15  
9  
11  
5  
7  
1  
3
```

Print all elements in reverse:

```
15  
13  
11  
9  
7  
5
```

```
3  
1  
  
Number of rows: 4  
  
Number of columns in row 0: 2  
  
Number of columns in row 1: 2  
  
Number of columns in row 2: 2  
  
Number of columns in row 3: 2  
  
First row: 1 3  
Last row: 13 15  
Number of rows: 4  
Number of columns: 2  
  
Second row: 5 7  
Largest value in the array: 15
```

Create a 3x3 2D array to store the numbers 1 to 9.

#Print only the diagonal elements.

```
public class Main {  
    public static void main(String[] args) {  
        // Create a 3x3 2D array with numbers 1 to 9  
        int[][] array = {  
            {1, 2, 3},  
            {4, 5, 6},  
            {7, 8, 9}  
        };  
  
        // Print only the diagonal elements (where row index == column  
index)  
        System.out.println("Diagonal elements:");
```

```

        for (int i = 0; i < array.length; i++) {
            System.out.println(array[i][i]);
        }
    }
}

```

Diagonal elements:

1
5
9

```

public class Main {
    public static void main(String[] args) {
        // Create a 3x3 2D array with numbers 1 to 9
        int[][] array = {
            {1, 2 },
            {4, 5 },
            {7, 8 }
        };

        // Print only the diagonal elements (where row index == column
index)
        System.out.println("Diagonal elements:");
        for (int i = 0; i < array.length; i++) {
            System.out.println(array[i][i]);
        }
    }
}

```

Wrong

Meya methana array[3][3] value eka illana nisa

Error : ArrayIndexOutOfBoundsException

◆ ArrayList

ArrayList කියන්නේ මොකක්ද?

ArrayList කියන්නේ Java programming language එකේ තියෙන class එකක්.

එකෙන් අපිට:

- එකම data type එකේ values තරමටම එකට තියාගන්න
- අවශ්‍ය විට අලුත් values add කරන්න, ඕනෑම නැති ඒවා remove කරන්න
- Array එකක් වගේ index හරහා access කරන්න

❖ Arrays වලට වඩා ArrayList ලා වෙනස්ම හේතුව මොනවාද?

- Arrays වල size එක fix වෙනවා
(දදා: `int[] nums = new int[5];` → මෙතන items 5 කින්ම සීමා)

- **ArrayList** එකක size එක **dynamic**

(දදා: `ArrayList<String> names = new ArrayList<>();` → ඔහු තරම් add කරන්න පූඩ්වන්)

අන්තටම **ArrayList** එකකට Java වල තියෙන *primitive data types* direct දාන්ත බැං, ඒ වෙනුවට ජ්‍යාගේ **wrapper classes** දාන්ත ඕනෑම.

```
ArrayList<Integer> numbers = new ArrayList<>();
```

```
ArrayList<String> names = new ArrayList<>();
```

```
ArrayList<Double> prices = new ArrayList<>();
```

```
ArrayList<Character> letters = new ArrayList<>();
```

```
ArrayList<Boolean> flags = new ArrayList<>();
```

```
ArrayList<Float> weights = new ArrayList<>();
```

```
ArrayList<Long> bigNumbers = new ArrayList<>();
```

```
ArrayList<Short> smallNumbers = new ArrayList<>();
```

```
ArrayList<Byte> bytes = new ArrayList<>();
```

* උදාහරණයක්:

```
import java.util.ArrayList;

public class Demo {
    public static void main(String[] args) {
        ArrayList<String> fruits = new ArrayList<>();

        fruits.add("Apple");
        fruits.add("Mango");
        fruits.add("Banana");
```

```

        System.out.println(fruits); // [Apple, Mango, Banana]

        fruits.remove("Mango");
        System.out.println(fruits); // [Apple, Banana]
    }
}

```

● ArrayList වල නියෙන ප්‍රධාන methods:

<u>Method</u>	<u>කරන්නේ මොකක්ද?</u>
<code>add(item)</code>	අලුත් item එකක් add කරනවා
<code>remove(item)</code>	item එක remove කරනවා
<code>get(index)</code>	index එකෙන් item එක ගන්නවා
<code>set(index, item)</code>	index එකේ නියෙන element එක වෙනස් කරනවා.
<code>size()</code>	item ගණන කියනවා
<code>clear()</code>	සියලුම item හිස් කරනවා

❖ **අැත්තවම ArrayList කියන්නේ** → "Normal array එකක් වගේ, නමුත් අවශ්‍ය
විට size එක අලුත් කරන, පහසුවෙන් handle කරන්න පූල්වන් class එකක්."

01. `add(item)`

ArrayList එකට අලුත් element එකක් add කරනවා.

```

import java.util.ArrayList;

public class Example1 {

    public static void main(String[] args) {

        ArrayList<String> fruits = new ArrayList<>();

```

```
fruits.add("Apple");

fruits.add("Mango");

System.out.println(fruits); // [Apple, Mango]

}

}
```

02. remove(item) නො remove(index)

Element එකක් remove කරනවා.

```
import java.util.ArrayList;

public class Example2 {

    public static void main(String[] args) {

        ArrayList<String> fruits = new ArrayList<>();

        fruits.add("Apple");

        fruits.add("Mango");

        fruits.add("Banana");

        fruits.remove("Mango"); // value එකක් remove කරනවා

        // fruits.remove(0); // index එකක් remove කරනවා
        // (Apple remove වෙනවා)

        System.out.println(fruits); // [Apple, Banana]

    }

}
```

```
}
```



03. get(index)

index එකක් විධානයෙන් element එකක් ගන්න.

```
import java.util.ArrayList;

public class Example3 {
    public static void main(String[] args) {
        ArrayList<String> fruits = new ArrayList<>();
        fruits.add("Apple");
        fruits.add("Mango");

        String firstFruit = fruits.get(0); // index 0 -> Apple
        System.out.println(firstFruit); // Apple
    }
}
```



04. set(index, item)

index එකේ නියෙන element එක වෙනස් කරනවා.

```
import java.util.ArrayList;

public class Example4 {
    public static void main(String[] args) {
```

```
ArrayList<String> fruits = new ArrayList<>();

fruits.add("Apple");

fruits.add("Mango");

fruits.set(1, "Banana"); // Mango ගෙනුවත් Banana

System.out.println(fruits); // [Apple, Banana]

}

}
```

05. size()

ArrayList හිලක් elements ගණන.

```
import java.util.ArrayList;

public class Example5 {

    public static void main(String[] args) {

        ArrayList<String> fruits = new ArrayList<>();

        fruits.add("Apple");

```

```
fruits.add("Mango");  
  
        System.out.println(fruits.size()); // 2  
    }  
}
```

06. clear()

ArrayList එක හිස් කරනවා.

```
import java.util.ArrayList;  
  
public class Example6 {  
  
    public static void main(String[] args) {  
  
        ArrayList<String> fruits = new ArrayList<>();  
  
        fruits.add("Apple");
```

```
fruits.add("Mango");  
  
fruits.clear();  
  
System.out.println(fruits); // []  
}  
}
```

◆ .equal

- `==` → memory එකේ එකම object එකද?
- `.equals()` → content එක සමානද?

මෙක paper එකේ ලියන්න:

Strings compare කරනකොට .equals() use කරන එක හොඳම practice එකයි,

`==` use කරාම හැමවිටම value එක නෙමේ, object identity බලනවා

➤ ==

- Memory address එක (reference) එකමද කියලා බලනවා.
- උදා:

```
String a = "hello";
String b = "hello";

System.out.println(a == b);           // true (string pool එකේ
එකම object එකට point කරන ත්‍යා)
```

➤ .equals()

- Strings වල actual value එක (content) එකමද බලනවා.
- උදා:

```
String a = new String("hello");
String b = new String("hello");

System.out.println(a == b);           // false (නව objects දෙකක්,
memory addresses වෙනස්)
System.out.println(a.equals(b)); // true (value එක "hello"
දෙකේම)
```

Habai default value eka widihata

String y == null;

Kiyala liyanna puluwan

◆ printf()

printf ලෙස ලියනවා) කියන්නේ C, C++, Java වගේ භාෂාවල පාවිච්චී කරන **formatted output function** එකක්.

Java වලද **printf** method එක තියෙනවා. ඒක **System.out** object එකේ method එකක්:

java

CopyEdit

```
System.out.printf("format string", arguments...);
```

එක C language වල printf වගේම නුගාක් සමානයි – format specifiers (%d, %f, %s වගේ) භාවිතා කරනවා.

 උදාහරණ 1: **simple text print කිරීම**

```
System.out.printf("Hello, World!");
```

Output:

Hello, World!

 උදාහරණ 2: **variable එකක් print කිරීම**

```
int age = 25;
```

```
System.out.printf("My age is %d", age);
```

Output:

My age is 25

 උදාහරණ 3: **float value print කිරීම**

```
double price = 99.99;  
  
System.out.printf("Price: %.2f", price);
```

Output:

```
Price: 99.99
```

`%.2f` කියන්නේ decimal places 2ක් විතරම print කරන්න කියන format එක.

 උදාහරණ 4: **multiple variables print කිරීම**

```
String name = "Nimal";  
  
int marks = 85;  
  
System.out.printf("Name: %s, Marks: %d", name, marks);
```

Output:

```
Name: Nimal, Marks: 85
```

 **Format specifiers (Java වලද C වල වගේම)**

Specifier	අදහස

%d	integer
%f	floating point
%.2f	float with 2 decimals
%s	string
%c	character

⚡ ජ්‍යාරාංශය

Java වල `printf` කියන්නේ:

```
System.out.printf("format", variables...);
```

- `printf` => print formatted text.
- variables print කරන්න `%d`, `%f`, `%s` වගේ placeholders use කරනවා

printf() ue karanawa length eka fix karala thiyanganna

Length fix

Use, printf()

இடையில் printf() குறித்துள்ள "fixed width table" என்று அழைக்கப்படும் படியின் பெருமை மீது இந்த வகையில் பல வகையான பாதிகள் உண்டு.

✓ printf Fixed Width Table Format படியின் பாதி

```
#include <stdio.h>

int main() {
    // Table Header
    printf("-----+-----+-----+\n");
    printf(" | %-10s | %-10s | %-10s | \n", "Item", "Price", "Qty");
    printf("-----+-----+-----+\n");

    // Table Rows
    printf(" | %-10s | %10.2f | %10d | \n", "Apple", 50.0, 10);
    printf(" | %-10s | %10.2f | %10d | \n", "Banana", 25.5, 5);
    printf(" | %-10s | %10.2f | %10d | \n", "Mango", 100.75, 3);

    // Bottom border
    printf("-----+-----+-----+\n");

    return 0;
}
```

🔍 கூடின் விதம்:

- `%10s` → right-align string (எங்கேயுள்ள `%10s` or `%-10s` for left-align)
- `%10.2f` → total width 10, 2 decimal places
- `%10d` → width 10 integer

Output එක (Visual View):

Item	Price	Qty
Apple	50.00	10
Banana	25.50	5
Mango	100.75	3

Use, while{}

```
while (id.length() < 21) {  
    id += " ";  
}  
while (name.length() < 20) {  
    name += " ";  
}
```

Constant variables

final use

```
final String [] x = {"A", "E", "I", "O", "U"};
```

Fix value use karanakota api final use karala constant variable ekak widihata use karanawa

ex;- sathiye daas 7, (fix value)

English hodiye akuru serama, (fix value)

Day -10

◆ **ALGORITHMS**

Yam kisii task ekak wisadanna api piyawaren piyawara anugamanaya karana instructions

a step-by-step set of instructions designed to solve a specific problem or accomplish a particular task.

Big O Notation

Bio Notation yanu , algorithmayaka sankirnathwaya hoo awakaasha sankeernathwaya wisthara kiriimata pariganaka widyaawe baawithaawana balawath mewalamak

Big O notation is a powerful tool used in computer science to describe the time complexity or space complexity of algorithms

Api prashne wisadana wisdana wisana widihatama thawa kenek e widihatama karanna ona kila ekak na,

e kiyanne prashnayak wisadana krama kihipayak thiyenawa

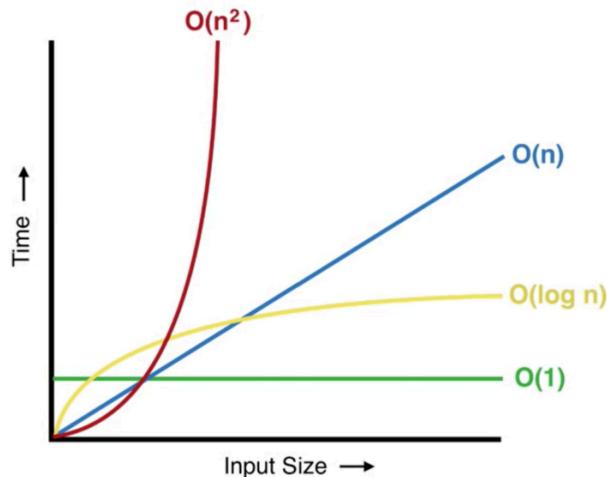
Prashnaya anuw ema prashnayata yana kaalaya wenas wenna puluhan,

Big o notation kiyanne aththatama label ekak

Eken pahadili karanawa eke time complexity eka haa space complexity eka

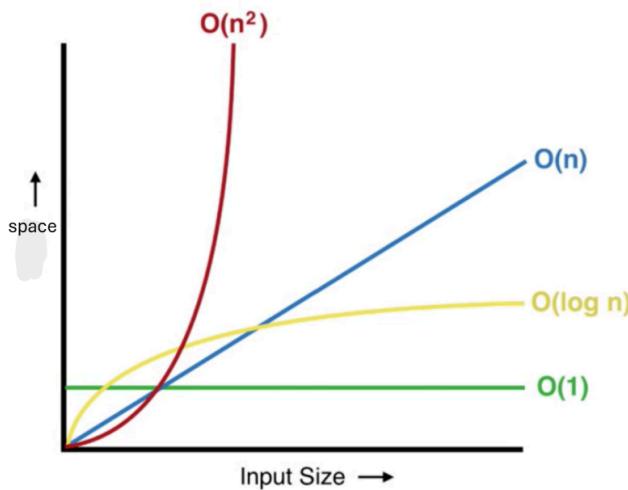
Time Complexity

The time complexity of an algorithm quantifies the amount of time taken by an algorithm to run as a function of the length of the input.



Space Complexity

The space complexity of an algorithm quantifies the amount of space taken by an algorithm to run as a function of the length of the input



- Input kochchra diila(input million ganak dunnath yanne ekama welawa) run karath eya ganna time eka ekama nam , eka api time complexity eka penwanne - **$O(1)$ space complexity eketh mema thathwedi me wagema thamai- eke complexity ekath $O(1)$**
- Input pramanaya wadi wanna wadi wanna ita sapekshawa eka karanna yanna wena kalaya wadi wenawa nam - eke time complexity eka, **$O(n)$ space complexity eketh mema thathwedi me wagema thamai- eke complexity ekath $O(n)$**
- Api dena addaana pramanaya wadi wanna wadi wanna eyala ganna kalaya wadi wenawa yam simawakata awm kochchara input gana wadi unath ethana idan ganna kaalaya ekamai ,
(input godak denakota meya gana kaalaya ekamai, kochchra dunnath wai wnne na)- me time complexity ekata kiyanawa - **$O(\log n)$ space complexity eketh mema thathwedi me wagema thamai- eke complexity ekath $O(\log n)$**
- Yam input pramanyak digna ynakota samanyayen inputa pramaneta anukulawa time eka wadi wenakota run wanna yana time eka ema pramanenma wai wenawa, tika tika input wadi wanna ganna kota yam seemaawakadi run wena time eka saahenna wadi wenawa nam - me time complexity kata kiyanawa - **$O(n^2)$ space complexity eketh mema thathwedi me wagema thamai- eke complexity ekath $O(n^2)$**

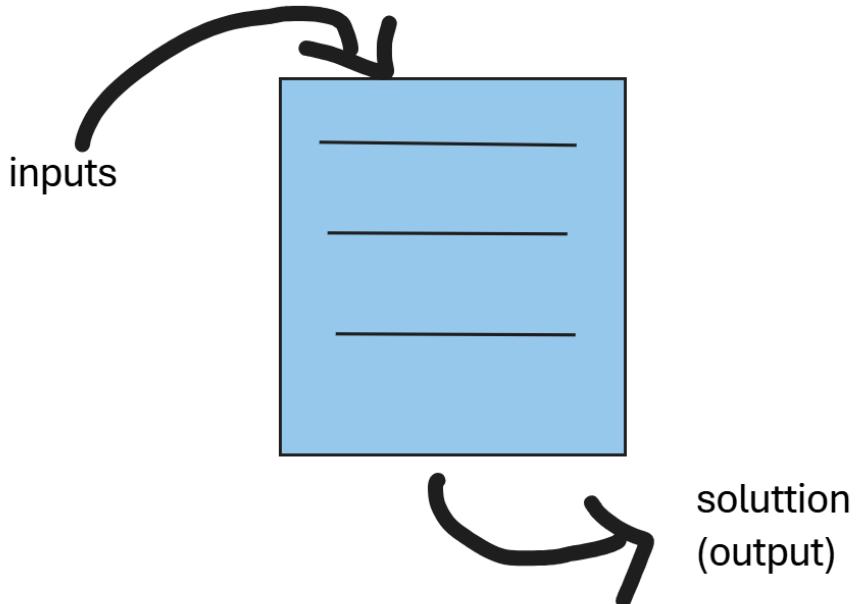
Code ek run wenakota memory eken ida pramanayk gannawa nisa memory eken ganna ida pramanayath e prashne anuwa thiranaya wenawa, eka nisa memory eka saha kalaya aduwen code karanna ona, mekata thamai **optimize** karla liyanawa kiyanne

Complexity 2k thiyenawa

- Time complexity
- Space complexity

Time Complexity

Api input karana dewal anuwah run wenna wena kaalaya thiranaya wenawa, Mekata api kiyanawa Time Complexity kiyala



The time complexity of an algorithm quantifies the amount of time taken by an algorithm to run as a function of the length of the input.

Yam kisii task ekak wisadanna api piyawaren piyawara anugamanaya karana instructions algorithms ,
Api dena input anuwa thamai solution eka enne
Memory eken gnna ida pramanaya gana salakilimath wiya yuhui

Space Complexity

Algorithamaya kriyathmaka wana wita ape wadeta kochchara idak gannawada kiyana eka memory eken

The space complexity of an algorithm quantifies the amount of space taken by an algorithm to run as a function of the length of the input.

අගුලෙන් තමය ක්රියාත්මක වන විට, එය භාවිතා කරන මතක ජ්‍රේමාණය ආදානනදේ දිගට අනුව මැනීමක් ලෙස the space complexity හැඳින්වේ.

Adaana diga = adaana pramaanaya

Sorting Algorithms

List ekak ho array ekak thula athi daththa arohana hoo awarohana piliwelakata sakas kiriimata baawithaa karana kramayaki

Sorting algorithms are methods used to rearrange a list or array of elements in a particular order, typically **ascending** or **descending**. Sorting is a fundamental concept in computer science that helps in organizing data for efficient searching, processing, and presentation.

Ex:- [4, 3 , 4,1, 2]

12345 (aarohana)

54321 (awarohana)

Sorting Algorithm jathi 3i,

- Insertion sort
- Bubble sort
- Selection sort

Class ekedi katha karanne Bubble sort eka gana witharai,
Viva ekedi ahana amaruma prashne meka wenna puluwan
Bubble sort muster karagana eka nisa

Bubble Sort

Viva ekedi ahana amaruma prashne meka wenna puluwan
Bubble sort muster karagana eka nisa

Bubble Sort is a simple sorting algorithm that repeatedly steps through the list, compares adjacent elements, and swaps them if they are in the wrong order. This process continues

Watch this for more references 

<https://youtu.be/Dv4qLJcxus8?si=lob3jpcf9HVkdJun>

```
class Main {  
    public static void bubbleSort(int[] array) {  
        for (int i = 0; i < array.length - 1; i++) {  
            for (int j = 0; j < array.length - i - 1; j++) {  
                if (array[j] > array[j + 1]) {  
                    int temp = array[j];  
                    array[j] = array[j + 1];  
                    array[j + 1] = temp;  
                }  
            }  
        }  
    }  
  
    public static void main(String[] args) {  
        int[] arr = {9, 10, 1, 3, 5, 2, 4, 6, 8, 7};  
        bubbleSort(arr);  
        System.out.println(Arrays.toString(arr));  
    }  
}
```

□ Bubble Sort කියන්නේ මොකද්ද?

Bubble sort කියන්නේ ඉතාමත් සරල sorting algorithm එකක්.
මෙහිදී list එකේ element දෙකක් එකින් එක එකට සම්බන්ධ කරලා, ඒවා
අනිවාර්ය පිළිවෙළින් (අදා: ascending / descending) නොමැති නම් ඒ දෙක
exchange (swap) කරනවා.

මේ ක්‍රියාව නැවත නැවත (repeatedly) කරනවා, list එකේ මුළු element විකම sorted වෙන නෙක්.

Aga idan mulata thamai meka sort karanne

උදාහරණයක්:

අපි ගනීමු list එක:

[5, 2, 9, 1, 5, 6]

Step 1:

- compare 5 & 2 → 5 > 2 → swap → [2, 5, 9, 1, 5, 6]
- compare 5 & 9 → 5 < 9 → no swap
- compare 9 & 1 → 9 > 1 → swap → [2, 5, 1, 9, 5, 6]
- compare 9 & 5 → 9 > 5 → swap → [2, 5, 1, 5, 9, 6]
- compare 9 & 6 → 9 > 6 → swap → [2, 5, 1, 5, 6, 9]

Step 2:

- compare 2 & 5 → 2 < 5 → no swap
- compare 5 & 1 → 5 > 1 → swap → [2, 1, 5, 5, 6, 9]
- compare 5 & 5 → equal → no swap
- compare 5 & 6 → 5 < 6 → no swap

Step 3:

- compare 2 & 1 → 2 > 1 → swap → [1, 2, 5, 5, 6, 9]

දැන් sorted!

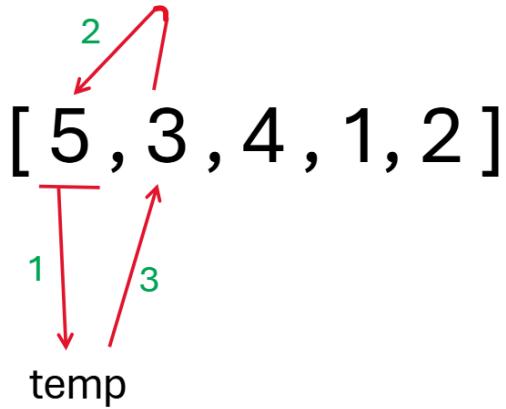
⌚ Bubble Sort විශේෂත්වය:

- ✓ ඉතාමත් සරලයි, හැමෝට්ටම නේරුම් ගන්න ලේසි
 - ✗ අඩුපරෘති (efficiency) – විශේෂයෙන් විශාල list වල
-

💻 Java වල Bubble Sort උදාහරණයක්:

```
import java.util.Arrays;

class Main {
    public static void main(String [] args) {
        int[] arr = {5, 2, 9, 1, 5, 6};
        for (int i = 0; i < arr.length - 1; i++) {
            for (int j = 0; j < arr.length - 1 - i; j++) {
                if (arr[j] > arr[j + 1]) {
                    // swap arr[j] and arr[j+1]
                    int temp = arr[j];
                    arr[j] = arr[j + 1];
                    arr[j + 1] = temp;
                }
            }
        }
        System.out.println(Arrays.toString(arr));
    }
}
```



```
import java.util.Arrays;

class Main {
    public static void main(String [] args) {
        int[] arr = {5, 3, 4, 1, 2};

        System.out.println(Arrays.toString(arr));

        for (int i = 0; i < arr.length; i++) {
            System.out.println(arr[i]);
        }
    }
}
```

```
import java.util.Arrays;

class Main {
    public static void main(String [] args) {
        int[] arr = {5, 3, 4, 1, 2};

        System.out.println(Arrays.toString(arr));

        for (int i = 0; i < arr.length -1; i++) {
/* api mehema -1 gahnne time complexiti wena nisa
           mokada eya anthima value ekath check karala thawa
parak loop wen nisa run wenna yana

```

```

        time eka wadi wenawa (ithamath podi pramanekin)
complexity eke case ekak thinawa , eka nisa api kiyanawa -1
dala anthima value eka check karanna epa
            kiyala (-1 danmath wada, time complexity thamai
case eka) */
// length eke antima value eka ai methana ganne naththe bubble
sort eka anthima ekedi use wenne nthi nisa anthima eka check
wenne ekata kalin ekath ekka , ekai ehema wenne
        System.out.println(arr[i]);
    }

}
}

```

```

import java.util.Arrays;

class Main {
    public static void main(String [] args) {
        int[] arr = {5, 3, 4, 1, 2};

        System.out.println(Arrays.toString(arr));

        for (int i = 0; i < arr.length -1; i++) {
            for (int j = 0; j < arr.length - 1 - i; j++) {
                System.out.println(arr[j]);
            }
        }
    }
}

```

```

import java.util.Arrays;

class Main {
    public static void main(String [] args) {
        int[] arr = {5, 3, 4, 1, 2};

        System.out.println(Arrays.toString(arr));

        for (int i = 0; i < arr.length -1; i++) {
            for (int j = 0; j < arr.length - 1 - i; j++) {
                System.out.print(arr[j]);
            }
        }
    }
}

```

```

        System.out.println();
    }

}

[5, 3, 4, 1, 2]
5341
534
53
5
Anthima kena wenakal yannenaththe anthima kena ceck wenne ita kalin kena ekka
nisa

```

```

import java.util.Arrays;

class Main {
    public static void main(String [] args) {
        int[] arr = {5, 3, 4, 1, 2};

        System.out.println(Arrays.toString(arr));

        for (int i = 0; i < arr.length -1; i++) {
            for (int j = 0; j < arr.length - 1 - i; j++) {
                //ehapathte kena aragena eya meyata wada
wadinamema
                // value eka temp ekata dagena array j ta array
j+1 ge value eka danna
                // array j+1 t tempge kenaaragena awilla daanna
                if (arr[j] > arr[j + 1]) {
                    int temp = arr[j];
                    arr[j] = arr[j + 1];
                    arr[j + 1] = temp;
                }

                System.out.print(arr[j]);
            }

            System.out.println();
        }
        System.out.println(Arrays.toString(arr));
    }
}

```

Aarohana piliwelata sort wenawa

```
import java.util.Arrays;

class Main {
    public static void main(String [] args) {
        int[] arr = {5, 3, 4, 1, 2};

        System.out.println(Arrays.toString(arr));

        for (int i = 0; i < arr.length -1; i++) {
            for (int j = 0; j < arr.length - 1 - i; j++) {
                if (arr[j] > arr[j + 1]) {
                    int temp = arr[j];
                    arr[j] = arr[j + 1];
                    arr[j + 1] = temp;
                }
            }
        }

        System.out.println(Arrays.toString(arr));
    }
}
```

Awarohana piliwela

```
import java.util.Arrays;

class Main {
    public static void main(String [] args) {
        int[] arr = {5, 3, 4, 1, 2};

        System.out.println(Arrays.toString(arr));

        for (int i = 0; i < arr.length -1 ; i++) {
            for (int j = 0; j < arr.length -1 - i; j++) {
                if (arr[j] < arr[j + 1]) { // methana > lakuna
maru karawapu gaman awarohana piliwelata hadenawa
                    int temp = arr[j];
                }
            }
        }

        System.out.println(Arrays.toString(arr));
    }
}
```

```

        arr[j] = arr[j + 1];
        arr[j + 1] = temp;
    }
}

}
System.out.println(Arrays.toString(arr));
}
}

```

```

import java.util.Arrays;

class Main {
    public static void main(String [] args) {
        int[] arr = {5, 3, 4, 1, 2};

        System.out.println(Arrays.toString(arr));

        for (int i = 0; i < arr.length -1; i++) /* api mehema
-1 gahnne time complexiti wena nisa
            mokada eya anthima value ekath check karala thawa
parak loop wen nisa run wanna yana
            time eka wadi wenawa (ithamath podi pramanekin)
complexity eke case ekak thinawa , eka nisa api kiyanawa -1
dala anthima value eka check karanna epa
            kiyal (-1 danmath wada, time complexity thamai
case eka) */
            for (int j = 0; j < arr.length -1 - i; j++) {
                if (arr[j] < arr[j + 1]) { // methana > lakuna
maru karawapu gaman awarohana piliwelata hadenawa
                    int temp = arr[j];
                    arr[j] = arr[j + 1];
                    arr[j + 1] = temp;
                }
            }
        }
        System.out.println(Arrays.toString(arr));
    }
}

```

```

import java.util.Arrays;

class Main {
    public static void main(String [] args) {
        int[] arr = {5, 3, 4, 1, 2};

        System.out.println(Arrays.toString(arr));

        for (int i = 0; i < arr.length -1; i++) {
            for (int j = 0; j < arr.length -1 - i; j++) {
                if (arr[j] > arr[j + 1]) { // ascending order
                    int temp = arr[j];
                    arr[j] = arr[j + 1];
                    arr[j + 1] = temp;
                }
            }
        }

        System.out.println(Arrays.toString(arr));
    }
}

```

Searching Algorithms

Array ekak athule menna me wage deyak thinawada kila hoyaganna searching algorithm use karanawa

Searching algorithms are essential tools in computer science used to locate specific items within a collection of data.

Searching algorithm dekak thiyenawa,

- Linear Search
- Binary Search (viva wala ahnna ida thinawa)

Time Complexity and Space Complexity for the above Algorithms

Algorithm	Best Case Time	Average Case Time	Worst Case Time	Space Complexity
Insertion Sort	$O(n)$	$O(n^2)$	$O(n^2)$	$O(1)$
Bubble Sort	$O(n)$	$O(n^2)$	$O(n^2)$	$O(1)$
Selection Sort	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(1)$
Linear Search	$O(1)$	$O(n)$	$O(n)$	$O(1)$
Binary Search	$O(1)$	$O(\log n)$	$O(\log n)$	$O(1)$

- Best case time- Algorithm eka wada karanna puluwan hodama widiha
- Average case time - Average
- Worst case time- Narakama widiha
- Space complexity- Spaceekath ekka wada karana widiha

Me 5 denagema input pramane wadi una kila wenas wenne na, input wadi una kila algorithm eka anuwa memory eken space ganna pramanaya samaanai

◆ CW Theory

◆ class

static

Java වල **static** කියන්නේ **class-level** keyword එකක්. එකෙන් කියවෙන්නේ **මෙම variable** එක (හෝ **method** එක) **object** එකකට සම්බන්ධ නොවේයි, ඒ වෙනුවට **class** එකටම සම්බන්ධ වේයි කියන එකයි

```
public class Main {  
    static String username = "amg";  
  
    public static void main(String[] args) {  
        System.out.println(Main.username); // "amg" print වේයි  
    }  
}
```

or

```
public class Main {  
    static String username = "amg";  
  
    public static void main(String[] args) {  
        System.out.println(username); // "amg" print වේයි  
    }  
}
```

මෙම **username** variable එක **class variable** එකක් (static variable) කියලා කියනවා.

📌 **static** variable එකක විශේෂත්වය:

1. **Class** එකට සම්බන්ධයි – object (method) එකකට නොවේ.
 - එනම් **Main** class එකේ object එකක් සාදලා variable එක access කරන්න one තැහැලි
 - **Main.username** කියලා access කරන්න පූඩ්වන් (දැන් **main** method එක **static** නිසා **username** direct විට access වෙලා තියෙනවා).
2. **Object** එකක් හෝ **instance** එකක් සැදුවට තනි **copy** එකක් තියෙන්නෙම තියෙනවා.
 - එකම variable එක shared වෙනවා class එකේ හැම method එකක් අතරම.

Ex:-

```
static Scanner input = new Scanner(System.in);
```

Method kihipayak thiyenawanam mehema danmama Scanner eka hama method ekatama share wenawa wena wenama daanna awashya na

```
static Scanner input = new Scanner(System.in);
```

මේකෙන් Scanner object එකක් (මල් example එකේ input) **class-level variable** එකක් විදිහට නිර්මාණය වෙනවා. ඒ කියන්නේ:

- ඒ Scanner object එක **class** එකටම සම්බන්ධයි, object එකකට නොවේයි.
- ඒ නීසා class එකේ තියෙන හැම static method එකකින්ම එකම input object එක use කරන්න පුළුවන්.
- නව object එකක් හඳාගන්න ඇවශය නැහැ මෙත්ම එකකට හැම වාරයකම.

 මේකෙන් ලැබෙන ප්‍රතිලාභ:

1. **Code duplication නැහැ** – හැම method එකකම new Scanner(System.in) කියලා දෙන්න ඕනෑම නැ.
2. **Memory save වෙනවා** – එකම Scanner object එක reuse කරන නීසා.
3. **Input stream conflict නැහැ** – System.in එකට Scanner එකක් විතරක් භාවිතා කිරීම best practice එකක්

!**Bonus Tip:**

- Program එකේ Scanner එක close කරනවා නම් (input.close()), ඒක **program** ඇවශානයේ කරන්න. නැත්තම් System.in stream එක close වෙලා error එනවා.

◆ login

nextLine();

- මෙහිදී space නිඩුණුන් ඒවාවන් capture කරයි.

```
System.out.print("\tUser Name : ");
```

```
String u = input.nextLine();
```

● මෙහිදී username එක input කරන්න user enter කරන name එක space නියෙන්න පූලවන්:

e.g., "praveen gunasekara"

එහෙම space නියෙන value එකක් input එකක් විදිහට accept කරන්න නම් nextLine() යොදීය යුතුයි.

next() use කරා නම් "praveen gunasekara" වලින් "praveen" විතරයි ගන්නවා, gunasekara discard වෙනවා.

එ නිසා ඔයාගේ code eke **nextLine()** කියන්නේ User එක type කරන සම්පූර්ණ line එක read කරන එකයි.

nextLine() method එක use කරන වාසියක් තමයි:

- ජුර්ණ line එක capture කරන්න පූලවන්.
- Spaces නිඩුණුන් ගැටුවක් නැහා

`input.nextLine();`

method එකට පෙර `nextInt()` වගේ method එකක් දාලා තියෙන්නේ නම්, `nextLine()` එක **empty line** එකක් capture කරන්න පූඩ්වන්. ඒක avoid කරන්න,
`input.nextLine();` එකක් extra දාන්න ඕනෑම sometimes.

```
Scanner input = new Scanner(System.in);
int age = input.nextInt();
String name = input.nextLine(); // මේක line එකක් skip වෙනවා!
```

◆ මෙහෙම `nextLine()` එක කුඩා input එකක්ම අරන් ඉවරයි, ඒ තිසා skip වෙන්න පූඩ්වන්. ඒක avoid කරන්න, `input.nextLine();` ekak extra danna ona:

```
int age = input.nextInt();
input.nextLine(); // consume the leftover \n
String name = input.nextLine(); // now this works correctly
```

අවුල තියෙන්නේ මෙහෙම `input mixing` එකේදී:

```
System.out.print("Enter age: ");
int age = input.nextInt(); // ➡ මේක numbers පමණක් read කරනවා
```

```
System.out.print("Enter name: ");
String name = input.nextLine(); // ❌ මේක skip වෙනවා!!
```

ଓ ඇයි skip වෙන්නේ ?

◆ `nextInt()` කියන එක read කරන්නේ number එක විතරයි – **enter/return key** එකෙන් එන ඩැලු (newline) එක කැමති වෙන්නේ නැහැ. එය `input buffer` එක් තියයි.

◆ Input Buffer කියන්නේ මොකද්ද?

► Input buffer කියන්නේ user එක keyboard එකෙන් type කරලා Enter key එක දාපු input එක, Java program එකට read කරගන්නට පෙර temporarily තියෙන memory එකක්

◆ පසුව තියන `nextLine()` එකට එය (empty line එකක්) දැක්කම එයා හිතන්නේ “අනේ! line එකක් already තිබුණා!” කියලා skip කරලා යනවා.

ಶೀಕ avoid ಕರನ್ನೆನು ಮೊಖೆಮಡಿ:

```
System.out.print("Enter age: ");
int age = input.nextInt();
input.nextLine(); // ⚡ ENTER key ಲಿಕೆನ್ ಆಗ್ `\\n` ಲಿಕ consume ಕರಲು discard ಕರನವು

System.out.print("Enter name: ");
String name = input.nextLine(); // ✅ ಇನ್ ಮೊಕ ವೈಟಿ ಕರನವು
```

Example full code:

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter your age: ");
        int age = input.nextInt();      // user enters 25 + presses Enter

        input.nextLine(); // ⚡ VERY IMPORTANT: clears the leftover '\\n'

        System.out.print("Enter your name: ");
        String name = input.nextLine(); // now this reads the name correctly

        System.out.println("Age: " + age);
        System.out.println("Name: " + name);
    }
}
```

- ◆ එකම data type එකෙන් `next()` එකක්ද `nextLine()` එකක්ද consecutively use කරේන්, ඒ අතර `input.nextLine();` එකක්දාන්න ඕනෑද?

ඉව්,
next() එකක් use කරලා පසුව `nextLine()` එකක් use කරනවනම්, අනිවාරයෙන්ම `input.nextLine();` එකක් මැදිහත් කරන්න ඕන.

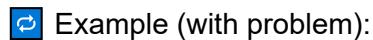


◆ `next()`:

- Stop reading at **space, tab, or newline**.
- Leaves the **newline character \n** in the input buffer.

◆ `nextLine()`:

- Read everything until it hits a newline (`\n`).
- If a newline is already there (from `next()`), it just reads an empty line!



```
Scanner input = new Scanner(System.in);
```

```
System.out.print("Enter first name: ");
String fname = input.next();      // e.g., "Praveen" ↵
```

```
System.out.print("Enter full line: ");
String line = input.nextLine();   // ❌ Skips! Read empty!
```

👉 **line** variable එකට empty string assign වෙනවා, මක්නිසාද `next()` එකෙන් ජය `\n` input buffer එක් තිබුණා.

Fixed Version:

```
Scanner input = new Scanner(System.in);

System.out.print("Enter first name: ");
String fname = input.next();      // "Praveen" ↴

input.nextLine();                //  discard leftover \n

System.out.print("Enter full line: ");
String line = input.nextLine();   //  now this works
```

Summary :

Sequence	Why?
next() → next()	<input checked="" type="checkbox"/> നാശി Both stop at space/newline
nextLine() → nextLine()	<input checked="" type="checkbox"/> നാശി Both consume full line
next() → nextLine()	<input checked="" type="checkbox"/> അവി next() leaves \n, nextLine() skips
nextInt() → nextLine()	<input checked="" type="checkbox"/> അവി Same reason — leftover newline remains

එකම data type එකෙන් input අහනව නම් (උදාහරණයක් විදියට nextLine() එකෙන් වරෙන් වර),
input.nextLine(); එක extra දාන්න ඕනෑම නැ.

 උදාහරණයක් – OK / නිවැරදි:

```
Scanner input = new Scanner(System.in);
```

```
System.out.print("Enter name: ");
String name = input.nextLine();
```

```
System.out.print("Enter address: ");
String address = input.nextLine();
```

● මෙතනදී දෙකටම nextLine() use කරන නිසා:

- input buffer එක clean.
- ENTER key properly handle ලැබනවා.
- **No need for extra input.nextLine();**



System.exit(0);

System eken exit wennna

`System.exit(0);` කියන්නේ Java program එක **turam end (terminate)** කරන්න යොදාගන්න special method එකක්

📌 සරල වගයෙන්:

`System.exit(0);` කියන්නේ "මට මෙතැනින් වැඩ ඉවරයි, JVM එකෙන් program එක close කරන්න!" කියන එකයි.

🧠 Syntax Breakdown:

`System.exit(statusCode);`

- `System` – Java's built-in class.
- `exit(int statusCode)` – JVM (Java Virtual Machine) එකට program එකක් **end** කරන්න කියන method එක.
- `statusCode` – Exit code එක (int එකක්):
 - `0` → **Normal termination** (successfully exited).
 - `Non-zero` (e.g., `1`, `-1`) → **Abnormal termination** (error occurred).

✓ Example:

```
public class Main {  
    public static void main(String[] args) {  
        System.out.println("Program starting...");  
  
        // Exit the program here  
        System.exit(0);  
  
        // This line won't run  
        System.out.println("This won't be printed");  
    }  
}
```

}

System.exit(1); මොකද්ද වෙන්නේ?

මේ line එක run වුනාම, Java program එක **turam** ම **terminate** වෙනවා (stop වෙනවා), **error** එකක් හේතුකරගෙන කියලා JVM එකට signal එකක් යවලා.

👉 වෙනස් System.exit(0) vs System.exit(1):

Exit Code	ඇරැත
System.exit(0);	✓ Normal exit – වැඩ නීමාවට පසු
System.exit(1);	✗ Abnormal exit – දේශයකි, වැරදි

System.exit(-1) = System.exit(1)

Dekenma wenne ekama de

💡 උදාහරණයක්:

```
public class Main {  
    public static void main(String[] args) {  
        System.out.println("Checking login...");  
        boolean valid = false;  
  
        if (!valid) {  
            System.out.println("Login failed!");  
            System.exit(1); // Error exit  
        }  
  
        System.out.println("Welcome!");  
    }  
}
```

📊 Comparison Table:

Code	Meaning	Use Case Example
0	✓ Normal exit	Success, no error

1, -1, etc.

✖ Abnormal exit (error)

Input invalid, system failure, login failed, etc

toUpperCase();

```
String s = "yes";
s = s.toUpperCase(); // "YES"
```

◆ මෙකෙන් string එකේ නැම අක්ෂරයක්ම uppercase (capital letter) එකක් වෙලා return වෙනවා.

◆ Example:

- "y" → "Y"
- "n" → "N"
- "Yes" → "YES"

◆ මෙක use කරන හෝතුව: user එකට 'y', 'Y', ' ', 'y' වගේ mix input type දෙන්න පූඩ්වන්.

toUpperCase() use කළේත් case එක ignore කරලා compare කරන්න පූඩ්වන්.

trim();

```
String s = "   Y   ";
s = s.trim(); // "Y"
```

◆ trim() method එකෙන් string එකේ ඇරණුම සහ අවසානයේ තියෙන space (white spaces) remove වෙනවා.

◆ Example:

- " Y" → "Y"
- "N " → "N"
- " Y " → "Y"

◆ මෙක use කරන හෝතුව: user එක mistake එකක් විදියට space දාන්න පූඩ්වන්. ඒක නිසා trim() කරන්නේ ඒ white spaces remove කරන එක

```
String again = input.nextLine().toUpperCase().trim();
```

- මෙකෙන් user එක " y " වගේ input එකක් දුන්නොත්:

- .trim() → "y"
- .toUpperCase() → "Y"

✓ සාරාංශය:

Method	කාර්යය	උදාහරණය
toUpperCase()	සියල්ල capital letters කරයි	"yes" → "YES"
trim()	මුද්‍රිතය & අගේ space ඉවත් කරයි	" Y " → "Y"

Example

```
System.out.print("Do you want to go home page (Y/N):");
String again = input.nextLine().toUpperCase().trim();
if (again.equals("Y")) {
    clearConsole();
    homepage();
}
if (again.equals("N")) {
    clearConsole();
    credentialManage();
}
```

◆ findSupplierIndexById

Duplicate Value Check

```
public static int findSupplierIndexById(String id) {  
    for (int i = 0; i < supplierCount; i++) {  
        if (supplierIds[i].equals(id)) {  
            return i;  
        }  
    }  
    return -1;  
}
```

මෙය supplierIds array එකක් තුළුන් id එකක් හොයන method එකක්.
(search method එකක්)

```
if (findSupplierIndexById(id) != -1) {  
    System.out.println("Supplier ID already exists. Try another.\n");  
    continue;  
}
```

 මෙක කොහොමද වැඩ කරන්නේ?
String id කියන එක method එකට පාරමිටර් එකක් — ඒ කියන්නේ user එකෙන්
හෝ වෙනත් code එකකින් යවන supplier ID එකක්.

for loop එකක් තියෙන්නේ 0 සිට supplierCount නෙක් (0 to supplierCount - 1).

`supplierIds[i].equals(id)`

► i වන index එකේ තියෙන supplier ID එක, ඔයා යවන id එකට සමානය කියලා
බලන එක.

සමානනම (equals true nam) → ඒ index එක return කරයි.

සමාන කිසිම එකක් නැත්තා → -1 return කරයි (supplier එක තියෙන්නේ නෑ
කියන එක)

◆ deleteSupplier

Delete Supplier

```
// Shift arrays left
for (int i = index; i < supplierCount - 1; i++) {
    supplierIds[i] = supplierIds[i+1];
    supplierNames[i] = supplierNames[i+1];
}
supplierCount--;
```

මෙය supplier එකක් delete කරන විට, ඒ supplier එක array එකෙන් ඉවත් කරන algorithm එකයි

supplierIds[] සහ supplierNames[] කියන parallel arrays දෙකෙන් supplier එක delete කරන එක.

එකක් delete කලම, array එකේ පසුව තියෙන values වික "එක පියවරක් වමට shift කරලා" previous එක override කරනවා.

එනැනින් පසුව supplierCount-- කරලා, total supplier count එක අඩු කරනවා.

🧠 උදාහරණයක්:

Imagine this:

```
supplierIds = {"S001", "S002", "S003", "S004"}  
supplierNames = {"ABC", "XYZ", "DEF", "PQR"}  
supplierCount = 4
```

User deletes S002. ലൈംഗിക് index = 1.

Loop ലൈംഗിക്:

```
i = 1:  
    supplierIds[1] = supplierIds[2]; // "S003"  
    supplierNames[1] = supplierNames[2]; // "DEF"  
  
i = 2:  
    supplierIds[2] = supplierIds[3]; // "S004"  
    supplierNames[2] = supplierNames[3]; // "PQR"
```

പിന്തും പാട്ടാണ്:

```
supplierIds = {"S001", "S003", "S004", "S004"}  
supplierNames = {"ABC", "DEF", "PQR", "PQR"}
```

ഉള്ളിൽ `supplierCount--` ലൈംഗിക്:

```
supplierCount = 3
```

So "S002" effectively removed.

‘

◆ viewSuppliers

View Suppliers

```
// print each supplier
for (int i = 0; i < supplierCount; i++) {
    String id = supplierIds[i];
    String name = supplierNames[i];
```

🔍 මෙම කොටසේන් වෙන්නේ මොකදේද?

මෙය supplier list එක print කරන loop එකේ ආරම්භයයි. මෙය supplier list එක supplierIds[] සහ supplierNames[] වලින් loop කරලා, supplier ID සහ නම දෙකම id සහ name වෙනස්කම්වලට assign කරනවා.

🧠 විස්තරයෙන්:

supplierCount → දැන් තියෙන supplier ගණන.

supplierIds[i] → i වන සහ supplier එකේ ID.

supplierNames[i] → i වන සහ supplier එකේ නම.

මෙවා දෙකම assign කරනවා String id සහ String name කියන temporary variables වලට, එතැනින් print කරන්න

◆ deleteCategory

```
// Shift arrays to delete
for (int i = index; i < categoryCount - 1; i++) {
    categoryCodes[i] = categoryCodes[i + 1];
    categoryName[i] = categoryName[i + 1];
```

```
}
```

```
categoryCount--;
```

◆ addItem

Check Supplier Number

- මේ loop එක valid supplier number එකක් ලැබෙන තුරු repeat වෙනවා.

```
int supplierIndex;
while (true) {
    System.out.print("\nEnter Supplier Number > ");
    try {
        supplierIndex = Integer.parseInt(input.nextLine().trim()) - 1;
        if (supplierIndex < 0 || supplierIndex >= supplierCount) {
            System.out.println("Invalid supplier number. Try again.");
        } else {
            break;
        }
    } catch (NumberFormatException e) {
        System.out.println("Please enter a valid number.");
    }
}
```

මේ කොට්ඨාස එකේ සිද්ධවෙන්නේ user එකක් valid (යනාර්ථ) supplier number එකක් අසින එක. ඒක while-loop එකක් තුළින් try-catch block එකක් යොදාගෙන කරනවා.

පහසුවෙන් පැහැදිලි කරන්නම්:

▣ **while (true):**

- මෙක නවන්වන තුරු loop එකක්. user valid input එකක් දෙන තුරු නැවත නැවත input අහනවා.
-

```
System.out.print("\nEnter Supplier Number > ");
```

- User එකට prompt එකක් දක්වනවා: "Supplier Number එක ඇතුළත් කරන්න".
-

```
12 supplierIndex = Integer.parseInt(input.nextLine().trim()) - 1;
```

- User input එක:
 - `input.nextLine()` – user එකට type කරන string එක ගන්නවා.
 - `.trim()` – ඉදිරිපස සහ අග්‍රේ white spaces ඉවත් කරනවා.
 - `Integer.parseInt(...)` – ඒ string එක integer එකක් ලෙස පරිවර්තනය කරනවා.
 - `- 1` – because array/අරය වල index 0න් පටන් යන නිස්‍ය. උදා: Supplier 1 කියන්නේ `supplierIndex = 0`.

```
✓ if (supplierIndex < 0 || supplierIndex >= supplierCount)
```

- මෙතැනින් බලනවා user දීන්න නූත්‍රු number එක validද කියලා.
 - `supplierCount` කියන්නේ ඇත්තටම නියෝග සුප්පලා ගණන.
 - ✗ Invalid** නම්: "Invalid supplier number. Try again." කියලා කියනවා.
 - ✓ Valid** නම්: `break`; – loop එකෙන් බිඳා පිටවෙනවා.
-

```
⚠ catch (NumberFormatException e)
```

- User එක number එකක් වෙනුවට අකුරු හෝ වැරදි format එකක් දුන්නොත් `Integer.parseInt()` එක error එකක් දමනවා.
 - ඒක catch කරලා "Please enter a valid number." කියලා friendly error message එකක් දක්වනවා.
-  `NumberFormatException` කියන්නේ මොකදීද?
- Java එක් `Integer.parseInt(String s)` වගේ method එකක් run කරනෙකාට, string එක integer එකක් නොවුනොත්, Java එක error එකක් (exception එකක්) throw කරනවා

try catch

```
try {
    // Error එකක් (exception එකක්) වෙන්න පූලවන් risky code
} catch (ExceptionType e) {
    // Error එකක් අල්ලලා (catch කරලා) මෙහිදී handle කරන්න
}
```

 මෙහිදී වෙන්නේ:

1. `try` block එක තුළ – user input එකක් integer එකක් කරලා print කරනවා.
2. user "123" type කරේන් → ඔක්කොම හරි, `catch` block එක run වෙන්නේ නැහැ.
3. user "abc" type කරේන් → error එකක් throw වෙනවා (`NumberFormatException`)
4. ඒ error එක `catch` block එකෙන් අල්ලනවා, program එක crash වෙන්නේ නැහැ

 උදාහරණයක්:

```
import java.util.Scanner;
```

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

```

Scanner input = new Scanner(System.in);

try {
    System.out.print("Enter a number: ");
    int number = Integer.parseInt(input.nextLine());
    System.out.println("You entered: " + number);
} catch (NumberFormatException e) {
    System.out.println("Invalid number! Please enter digits only.");
}
}
}

```

 Summary Table:

කොටස	කාර්යය
try block	Error එකක් වෙන්න පුළුවන් code ලියන තැනු
catch block	Error එකක් වෙලාද බලපා, එක handle කරන තැනු
ExceptionType	Handle කරන්න යන error එකේ වර්ගය (e.g. NumberFormatException)
e	Error object එක – error message එක හෝ cause එක ගන්න පුළුවන්

* සරල වචන වලින්:

try-catch කියන්නේ "මට මෙහිදී දෝශයක් වෙන්න පුළුවන්, ඒ නිසා අලුතෙන් safe එකක් ගනීම්" කියන විදිහට Java එකට කියන කමයක්.

Printf

```
System.out.printf(" | %-7s | %-11s | %-16s | %-11s | %-8s |\n", itemSid[j], itemCodes[j],
itemDescription[j], itemPrices[j], itemQty[j]);
```

මෙක formatted output එකක්. එහෙම කියන්නේ ඇතුළත් කරන දත්තවල් තීරු (columns) වලට පෙළ ගැස්වලා, table format එකක print කරන එක.

🔍 printf() කියන්නේ මොකද්ද?

- printf() = print formatted
- ඔබට output එක control කරන්න දෙන විශේෂ method එකක්.
- % symbol එකකින් පෙන්වයි data type/placeholders.
- %-Ns කියන එකක් column width set කරනවා.

🧠 කොටස් වෙන්වෙන්ට පැහැදිලි කරමු:

	%-7s		%-11s		%-16s		%-11s		%-8s	
↑		↑		↑		↑		↑		↑
SID	CODE	DESC	PRICE	QTY						

📋 % format explained:

Code අදහස

- | | |
|--------|--|
| %s | String value එකක් print කරන්න |
| %d, %f | Integer/Decimal values print කරන්න |
| %-7s | String එක 7 character විතර පළවෙනි පැන්තට align කරලා print කරන්න (left-aligned) |
| %-11s | 11 character width එකක් |
| %-16s | 16 character width එකක් |

◆ rankItemPerUnitPrice

Bubble Sort

Sort array elements

```
// Simple bubble sort: sort by price descending
for (int i = 0; i < itemCount - 1; i++) {
    for (int j = 0; j < itemCount - i - 1; j++) {
        if (pricesCopy[j] < pricesCopy[j + 1]) {
            // swap prices

            double tempPrice = pricesCopy[j];
            pricesCopy[j] = pricesCopy[j + 1];
            pricesCopy[j + 1] = tempPrice;

            // swap codes
        }
    }
}
```

```

        String tempCode = codesCopy[j];
        codesCopy[j] = codesCopy[j + 1];
        codesCopy[j + 1] = tempCode;

        // swap sid
        String tmpSid = sidCopy[j];
        sidCopy[j] = sidCopy[j + 1];
        sidCopy[j + 1] = tmpSid;

        // swap description
        String tmpDesc = desCopy[j];
        desCopy[j] = desCopy[j + 1];
        desCopy[j + 1] = tmpDesc;

        // swap qty
        int tmpQty = qtyCopy[j];
        qtyCopy[j] = qtyCopy[j + 1];
        qtyCopy[j + 1] = tmpQty;

        // swap category names
        String tempName = namesCopy[j];
        namesCopy[j] = namesCopy[j + 1];
        namesCopy[j + 1] = tempName;
    }
}
}

```

Buble sort

```

class Main {
    public static void main(String [] args) {
        int[] arr = {5, 3, 4, 1, 2};

        System.out.println(Arrays.toString(arr));

        for (int i = 0; i < arr.length -1; i++) {
            for (int j = 0; j < arr.length -1 - i; j++) {
                if (arr[j] > arr[j + 1]) { // ascending order
                    int temp = arr[j];
                    arr[j] = arr[j + 1];
                    arr[j + 1] = temp;
                }
            }
        }
    }
}

```

```

        }

    }

    System.out.println(Arrays.toString(arr));
}

}

```

nextInt(n)

Generates a random integer between 0 (inclusive) and n (exclusive) → that means 0 to n-1.

```

import java.util.Random;

class Main {
    public static void main(String[] args) {
        Random r = new Random();
        int x = r.nextInt(5);
        System.out.println(x); // can be 0,1,2,3, or 4
    }
}

```

how **Math.random()** differs from **Random.nextInt(n)**?

◆ 1. Random.nextInt(n)

- Belongs to the **java.util.Random** class.
- Generates an **int** between **0** (inclusive) and **n** (exclusive).
- Example:

```

Random r = new Random();
int x = r.nextInt(10); // 0–9

```

- Thread-safe nemei, e nisa multiple threads ekak use karama unpredictable results enne.

◆ 2. `Math.random()`

- Static method in the `java.lang.Math` class.
- Returns a `double` between `0.0` (inclusive) and `1.0` (exclusive).
- If you want an integer, you must multiply and cast:

```
int y = (int)(Math.random() * 10); // 0–9
```

- Simpler for small tasks, no need to create a `Random` object.
-

⚖️ Summary

- `Random.nextInt(n)` → Directly gives you `int` in range `0..n-1`.
- `Math.random()` → Gives you `double 0.0..1.0`, must scale & cast if you want integers.

👉 Exam tip: If question says "Random method" → answer is always `nextInt(n)`

Feature	<code>Random.nextInt(n)</code>	<code>Math.random()</code>
Class	<code>java.util.Random</code>	<code>java.lang.Math</code>
Return type	<code>int</code>	<code>double</code>
Range	<code>0 to n-1</code>	<code>0.0 (inclusive) to 1.0 (exclusive)</code>

Usage	<code>r.nextInt(10); // 0–9</code>	<code>(int)(Math.random() * 10); // 0–9</code>
-------	------------------------------------	--

Need object?	<input checked="" type="checkbox"/> Yes (create new <code>Random()</code>)	<input type="checkbox"/> No (static method)
--------------	---	---

Flexibility	More methods available (<code>nextDouble()</code> , <code>nextBoolean()</code> , etc.)	Only one method
-------------	--	-----------------

Best for	Controlled, repeatable randomness	Quick, simple random values
----------	-----------------------------------	-----------------------------

👉 Easy memory tip:

- `Random.nextInt(n)` → int ready-made
- `Math.random()` → double, you must scale & cast

method converts a String to integer

1. `Integer.parseInt(String s)`

- Converts a String to a primitive `int`.

Example:

```
String str = "123";  
  
int num = Integer.parseInt(str); // num = 123
```

2. `Integer.valueOf(String s)`

- Converts a `String` to an `Integer` object (wrapper class).

Example:

```
String str = "123";  
  
Integer numObj = Integer.valueOf(str); // numObj = 123 (Integer)
```

- If you need a primitive `int`, Java will auto-unbox it:

```
int num = Integer.valueOf(str); // auto-unboxed to int
```

AMG-NOTE

JAVA *Object Oriented*

Programming

[JAVA OOP link](#)

Console Clear Method

```
class Example {  
    public static void main (String args[]){  
        int num=103;  
        if(num>0){  
            System.out.println (num+" is positive number");  
        }else if (num<0){  
            System.out.println (num+" is negative number");  
        }else{  
            System.out.println (num+"is 0");  
        }  
    }  
}  
//103 is positive number
```

```
import java.util.Scanner;
class Example{
    public static void main(String args[]){
        Scanner input = new Scanner(System.in);

        int num=input.nextInt();

        if(num>0){
            System.out.println(num+" is positive number");
        }else if(num<0){
            System.out.println(num+" is negative number");
        }else{
            System.out.println(num+" is 0");
        }
    }
}
```

```
import java.util.Scanner;
class Main{
    public static void main(String args[]) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter Your marks: ");
        int mark = input.nextInt();
        if (mark>=85) {
            System.out.println("A+ Grade");
        }else if(mark>=75){
            System.out.println("A Grade");
        } else if (mark>=65){
            System.out.println("B Grade");
        } else if (mark>=50){
            System.out.println("C Grade");
        } else if (mark>=35) {
            System.out.println("S Grade");
        }else {
            System.out.println("F Grade");
        }
    }
}
```

```
}
```

```
class Example{
    public static void main(String args[]){
        int b = 5;
        int c = b << 3; // 5*2^3
        System.out.println(c);
    }
}
```

```
class Example{
    public static void main(String args[]){
        int b = 5;
        int c = b >> 3; // 5/2^3
        System.out.println(c);
    }
}
```

PRF Remedial

```
import java.util.Scanner;
class Example {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("your name : ");
        String name = input.next(); // eka wachanayai
        System.out.println("hi "+name);

        System.out.print("your Location : ");
        String Location = input.next(); // eka wachanayai
        System.out.println(Location);

        System.out.println("Your Bill Amount :");
        double x = input.nextDouble();
```

```
        if (x>3000){  
            System.out.println("you have offer");  
        }else {  
            System.out.println("you haven't offer");  
  
        }  
    }  
}
```

```
import java.util.Scanner;  
class Example {  
    public static void main(String[] args) {  
        Scanner input = new Scanner(System.in);  
  
        System.out.print("your name : ");  
        String name = input.nextLine(); // eka line ekak liyanna puluwan  
        System.out.println("hi "+name);  
  
  
  
        System.out.print("your Location : ");  
        String Location = input.nextLine(); // eka line ekak liyanna  
        puluwan  
        System.out.println(Location);  
  
  
  
        System.out.println("Your Bill Amount :");  
        double x = input.nextDouble();  
  
  
        if (x>3000){  
            System.out.println("you have offer");  
        }else {  
            System.out.println("you haven't offer");  
  
        }  
    }  
}
```

```

import java.util.Scanner;

class Example {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("r: ");
        double r = input.nextDouble();

        double c = 2 * 3.14 * r;
        System.out.println("parimithiya: " + c);

        double A = 3.14 * r * r;
        System.out.println("wargapalaya: " + A);
    }
}

```

r: 7

parimithiya: 43.96

wargapalaya: 153.86

```

import java.util.Scanner;

class Example {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("r: ");
        double r = input.nextDouble();

        final double c= 2 * 3.14 * r; // passe wenas karanna bari wenna ode
        eka sthawara karanna final dama
        System.out.println("parimithiya: " + c);

        final double A = 3.14 * r * r;
        System.out.println("wargapalaya: " + A);
    }
}

```

```

import java.util.Scanner;
class Example {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.println("Simple Calculator");

        System.out.print("Enter first number: ");
        double num1 = input.nextDouble();

        System.out.print("Enter an operator (+, -, *, /): ");
        String operator = input.next();
    }
}

```

```

System.out.print("Enter second number: ");
double num2 = input.nextDouble();

switch (operator) {
    case "+"->System.out.println(num1 + num2);

    case "-">System.out.println(num1 - num2);

    case "*">System.out.println(num1 * num2);

    case "/">System.out.println(num1 / num2);

    default->System.out.println("Error: Invalid operator.");
}

}
}

```

```

import java.util.Scanner;

class Example {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.println("Simple Calculator");
        System.out.print("Enter first number: ");
        double num1 = input.nextDouble();

        System.out.print("Enter an operator (+, -, *, /): ");
        char operator = input.next().charAt(0);

        System.out.print("Enter second number: ");
        double num2 = input.nextDouble();

        switch (operator) {
            case '+':
                System.out.println(num1 + num2);

```

```

        break;

    case '-':
        System.out.println(num1 - num2);
        break;

    case '*':
        System.out.println( num1 * num2);
        break;

    case '/':
        if (num2 != 0) {
            System.out.println(num1 / num2);

        } else {
            System.out.println("Error: Division by zero!");
        }
        break;
    default: // default eka thiyyenne me cases walata adala nowana
dewal input karama defalt value eka thmai print karanne
        System.out.println("Error: Invalid operator.");
    }

}

```

```

import java.util.Scanner;

class Example {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.println("Simple Calculator");
        System.out.print("Enter first number: ");
        double num1 = input.nextDouble();

        System.out.print("Enter an operator ");
        String operator = input.next();
    }
}

```

```

System.out.print("Enter second number: ");
double num2 = input.nextDouble();

switch (operator) {
    case "+":
        System.out.println(num1 + num2);
        break;

    case "-":
        System.out.println(num1 - num2);
        break;

    case "*":
        System.out.println( num1 * num2);
        break;

    case "/":
        if (num2 != 0) {
            System.out.println(num1 / num2);

        } else {
            System.out.println("Error: Division by zero!");
        }
        break;
    default:
        System.out.println("Error: Invalid operator.");
}
}

```

Yam kisi agayak thawath agayakin beduwama e beduwama ena uththare purna sankyathmaka agaya saha ithuru agaya ena widihata program ekak liyanna

```
import java.util.Scanner;
```

```
class Example {
```

```
public static void main(String[] args) {  
    Scanner input = new Scanner(System.in);  
  
    System.out.print("Enter value : ");  
    int x = input.nextInt();  
  
    System.out.print("Enter value : ");  
    int y = input.nextInt();  
  
    if (y!=0) {  
  
        int z = x / y;  
        System.out.println("beduwama ena uththare :" + z);  
  
        int s = x % y;  
        System.out.println("ithuru gaana:" + s);  
    } else {  
        System.out.println("Error");  
    }  
}  
}
```

test

```
////////// IJSE_TEAM ///////////
```

```
import java.util.Scanner;

class Main {
    public static void password1() {
        System.out.println("Mr.Prasad Waduge");
        System.out.println("CEO | IJSE ");
    }

    public static void password2() {
        System.out.println("Mr.Yasendra Darshana");
        System.out.println("Senior manager & Lecturer | IJSE");
    }

    public static void password3() {
```

```
        System.out.println("Mr.Kavindu Akash");
        System.out.println("Senior Lecturer | IJSE");
    }

    public static void password4() {
        System.out.println("Mr.Dasun Srimal Athukorala");
        System.out.println("Senior Lecturer | IJSE");
    }

    public static void password5() {
        System.out.println("Mr.Anjana Thrishakya");
        System.out.println("Senior Lecturer | IJSE");
    }

    public static void line() {
        System.out.println("-----");
    }

    public static void admin() {
        System.out.println("IJSE");
    }

    public static void main(String args[]) {
        Scanner input = new Scanner(System.in);

        System.out.println();
        System.out.println("IJSE Institute of Software
engineering");
        System.out.println("      Team Details");
        line();
        System.out.println("      Login      ");
        line();

        System.out.print("User name : ");
        String name = input.next();

        System.out.print("Password : ");
        String password = input.next();

        switch (name) {
            case "prasad":
                line();

```

```

        if (password.equals("benz1234")) {
//.equals() කියන්නේ Java තං strings (එහෙම නැත්තම් objects)
compare කරන්න use කරන method එකක්.
            System.out.println("Hello " + name + "\n");
            password1();
        } else {
            System.out.println("Invalid Password");
        }
        break;

case "yasendra":
    line();
    if (password.equals("benz1234")) {
        System.out.println("Hello " + name + "\n");
        password2();
    } else {
        System.out.println("Invalid Password");
    }
    break;

case "kavindu":
    line();
    if (password.equals("benz1234")) {
        System.out.println("Hello " + name + "\n");
        password3();
    } else {
        System.out.println("Invalid Password");
    }
    break;

case "dasun":
    line();
    if (password.equals("benz1234")) {
        System.out.println("Hello " + name + "\n");
        password4();
    } else {
        System.out.println("Invalid Password");
    }
    break;

case "anjana":
    line();
    if (password.equals("benz1234")) {
        System.out.println("Hello " + name + "\n");

```

```
        password5();
    } else {
        System.out.println("Invalid Password");
    }
    break;

default:
    line();
    System.out.println("Unknown user.");
}

input.close();
}
}
```

```
class Example{
    public static void main(String args[]){
        byte x = 1;
        System.out.println(x); // prints 10
        System.out.println(~x); // prints -11      -(x+1)
    }
}
```

```

class Example {
    public static void main(String args[]) {
        int x = 10, y = 20, z = 30;
        int max;

        max = (x > y) ? ((x > z) ? x : z) : ((y > z) ? y : z);

        System.out.println("Max : " + max); // Max : 30
    }
}

```

Max : 30

Song lyrics

```

public class Main {

    // Sample lyrics
    static String[] lyrics = {
        "සිහිනේ ජායා පාරාලා",
        "ල් සිනා තුවෙ ඒ",
        "සිලිලේ සේයා මා කීවා",
        "තැසේනා සේ දරා වේදනා . . .",
        "\n\nහිත පාරන දේද වේදනා සන්තානේ",
        "මට නැසුණ දේද කිවේ මේ මන්දාරේ",
        "ඇපි ආ ගිය පාල මාවනේ මං ආවේ",
        "තැවනී මම පාර පිරුවා හන්තානේ //"
    };
}

```

```
// Start time delays (in milliseconds from program start)
static long[] startTimes = {
    1000, // 1 second delay
    3000, // 3 seconds
    5000, // 5 seconds
    7000, // 7 seconds
    9000, // 9 seconds
    11000, // 11 seconds
    13000, // 13 seconds
    15000 // 15 seconds
};

};

static final String[] LIGHT_COLORS = {
    "\u001B[94m", // Light Blue
    "\u001B[92m", // Light Green
    "\u001B[95m" // Light Magenta
};

// ANSI reset
static final String RESET = "\u001B[0m";

// Delay per character
static final int perCharDelayMs = 75;

public static void main(String[] args) throws
InterruptedException {
    long start = System.currentTimeMillis();

    for (int i = 0; i < lyrics.length; i++) {
        long now = System.currentTimeMillis();
        long waitTime = start + startTimes[i] - now;

        if (waitTime > 0) {
            Thread.sleep(waitTime);
        }

        String color = LIGHT_COLORS[i %
LIGHT_COLORS.length];
        printLineSmooth(lyrics[i], color);
    }
}
```

```
public static void printLineSmooth(String line, String color) throws InterruptedException {
    System.out.print(color);
    for (char ch : line.toCharArray()) {
        System.out.print(ch);
        Thread.sleep(perCharDelayMs);
    }
    System.out.println(RESET);
}
```

සිහිනේ ජායා පාරාලා
ප්‍රේ සිනා තුමේ ප්‍රේ
සිලිලේ සේයා මා කීවා
නැසෙනා සේ දරා වේදනා

හිත පාරන දේද වේදනා සන්නානේ
මට නයුණ දේද කීවේ මේ මන්දාරේ
අප්‍රා ආ ගිය පාල මාවනේ මං ආවේ
නැවත් මම පාර පිරුවා හන්නානේ //

