
SIS
ACADEMY
EQUINE technologies
Let's Get Started
Java Fundamental

Requirement Software

- JDK 1.8
- Intellij Idea 2018 or Latest
- Maven
- Git
- PostgreSQL 10

Day 01

- Create New Project
- Naming Convention
- Java Memory Model
- Git
- Demo Case

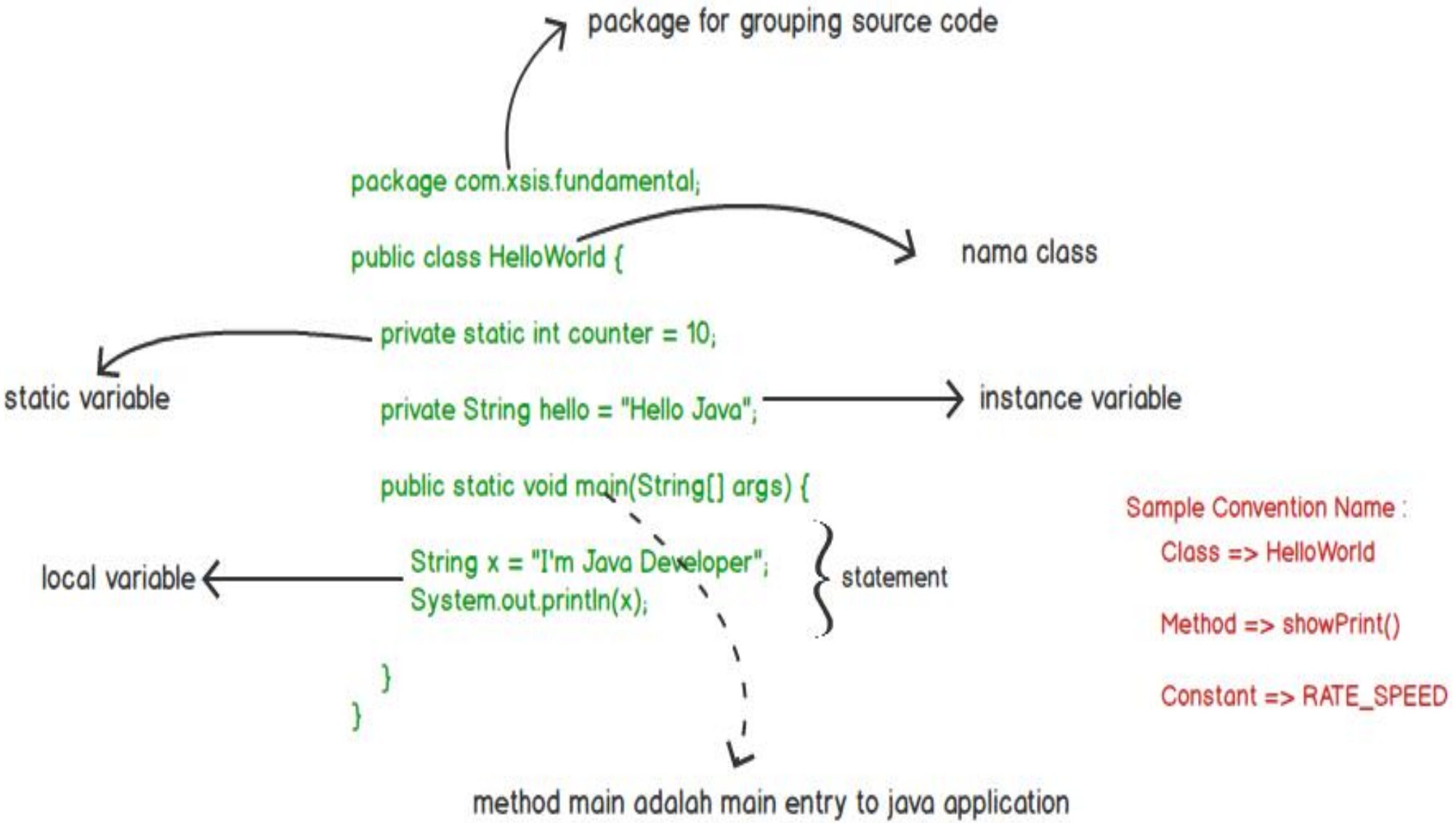
See video tutorial….

1. Hello World Project

```java
package com.xsis.fundamental;

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

See video tutorial….

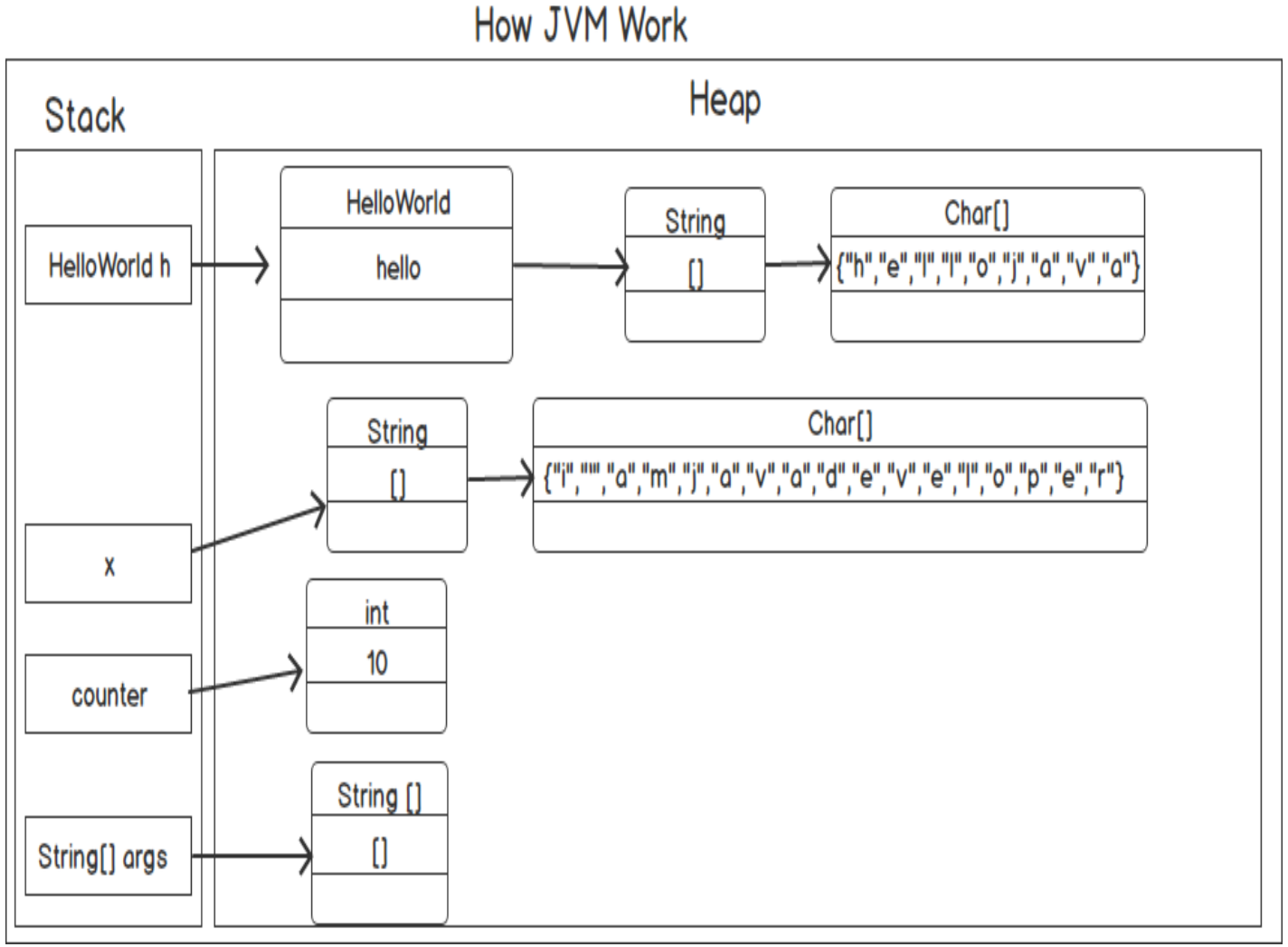
1. Naming Convention



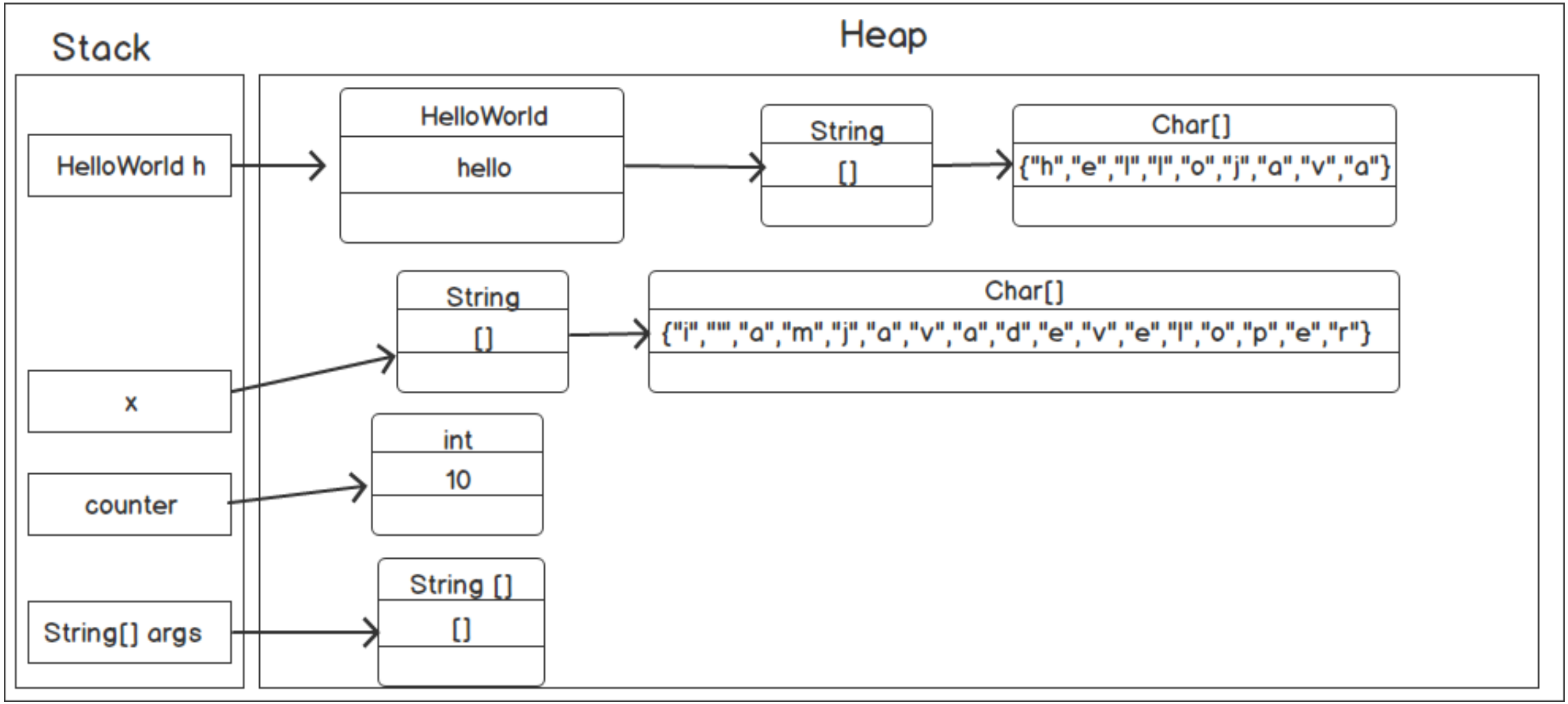
1. How JVM Work

Java compiler will read from top to bottom

```java
package com.xsis.fundamental;

public class HelloWorld {

    private static int counter =10;
    private String hello = "Hello Java";

    public static void main(String[] args) {
        String x = "I'm Java Developer";
        System.out.println(x);

        // call static variable
        System.out.println("static counter : "+counter);

        // call instance variable hello
        HelloWorld helloWorld = new HelloWorld();
        System.out.println(helloWorld.hello);
/*
        System.out.println("Hello World");
        System.out.println("Hello Java Xsis Programmer");
*/
    }
}
```



How JVM Work


Stack
Heap
HelloWorld h
x
counter
String[] args
HelloWorld
hello
String
[]
Char[]
{"h","e","l","l","o","j","a","v","a"}
String
[]
Char[]
{"i","","a","m","j","a","v","a","d","e","v","e","l","o","p","e","r"}
int
10
String []
[]

1. Using Git Locally

- git init

  Inisialisasi git di folder project

- git add .

  Menambahkan file-file baru ke repository git

- git commit –am “<comment>”

  Menambahkan file-file baru ke repository git

- git status

  Tampilkan perubahan files status project

- git checkout –b <branch-name>

  Create branch baru

- git checkout <branch-name>

  Pindah ke branch-name lain

- git branch –d <branch-name>

  Hapus branch-name lain

- git branch –m <branch-name>

  Ubah nama branch-name

- git branch –m <old-name> <new-name>

  Ubah nama branch-name current dengan nama

- git merge <branch-name>

  Merge current branch dengan branch-name lain

See video tutorial….

Case #1.1

Sebuah mobil menempuh jarak 110 kilometer dalam waktu 2 jam. Hitunglah kecepatan rata-rata mobil tersebut. (Petunjuk: kecepatan rata-rata = jaraktempuh/waktu tempuh). Kecepatan rata-rata = 110/2 = 55 km/jam.

Rumus :

v = s / t

- V = kecepatan (km/jam)
- S = jarak (km)
- t = waktu tempuh (jam)

```java
package com.xsis.fundamental.day01;

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

        int s = 110;
        int t =2;
        int v = 0;

        v = s/t;

        System.out.println("Kecepatan mobil : "+v+" KM/jam");
    }
}
```

Case #1.2 ~ Static Method

Rumus :

v = s / t

- v = kecepatan (km/jam)
- s = jarak (km)
- t = waktu tempuh (jam)

Create Method for each parameter!!!

```
public static int getJarak(){
    return ...
}

public static int getSpeed(){
    return ...
}

public static int getHour(){
    return ...
}
```

Static method hanya digunakan untuk scope class variable, hindari penggunaan method static, karena selain boros memory, juga code kita akan lebih mendekati procedural programming dibanding object oriented programming (oop)

Case #1.3 ~ Non Static Method With Param

Rumus :

v = s / t

- v = kecepatan (km/jam)
- s = jarak (km)
- t = waktu tempuh (jam)

Create Method for each variable rumus!!!

```
public int getJarak(int v, int t){
    return ...
}

public int getSpeed(int s, int t){
    return ...
}

public int getHour(int v, int s){
    return ...
}
```

Exercise #1

Seorang anak sedang mendorong benda dengan gaya sebesar 80 N,sehingga benda yang didorong tersebut bergerak dengan kecepatan tertentu.Bila suatu massa benda itu 8 kg,maka carilah percepatan benda tersebut!

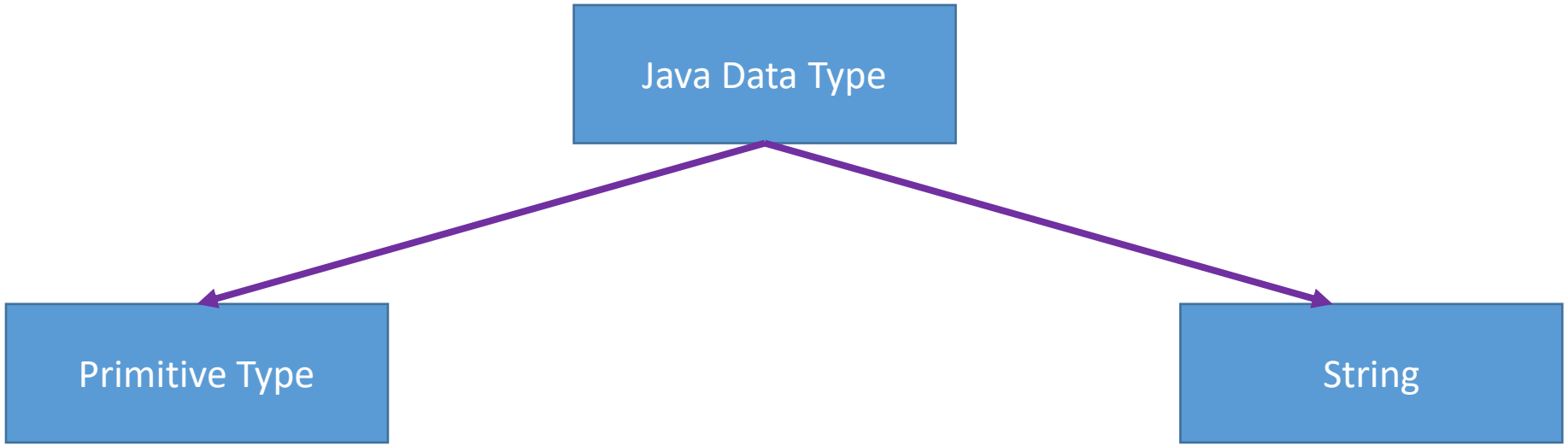
Rumus :

$$f = m * a$$

- f = Resultan gaya
- m = massa
- a = percepatan

Create dengan menggunakan non static method with parameter untuk tiap variable rumus, output yang dihasilkan dengan call method masing-masing di main() method .

2. Java Primitive Type



2. Data Type ~ Java Primitive Type

Type	Size	Range	Default*
boolean	1 bit	true or false	false
byte	8 bits	[-128, 127]	0
short	16 bits	[-32,768, 32,767]	0
char	16 bits	['\u0000', '\uffff'] or [0, 65535]	'\u0000'
int	32 bits	[-2,147,483,648 to 2,147,483,647]	0
long	64 bits	$[-2^{63}, 2^{63}-1]$	0
float	32 bits	32-bit IEEE 754 floating-point	0.0
double	64 bits	64-bit IEEE 754 floating-point	0.0

Note : Meskipun java sepenuhnya object oriented programming (OOP), karena alasan performansi, java masih menggunakan primitive type untuk tipe data di table atas

2. Data Type ~ Example

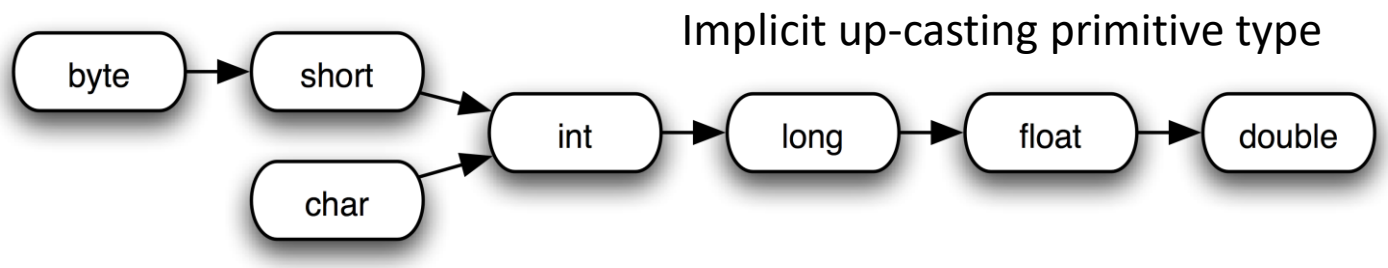
```
boolean bln = true; // booleans can only be 'true' or 'false'

byte b = 0x20; // using hexadecimal notation

short s = 500; // small integer

char c = 'A'; // must use single quotes to denote characters

char tab = '\t'; // other specials: \n, \r, \f, \b, \\, \', \"

int i = 1000000; // decimal notation

int j = 0x3FA0B3; // hexadecimal notation

int k = 0777; // octal notation

float f = 1.5f; // ‘f’ dibedakan untuk nilai double

long l = 2000000L; // 'L' dibedakan untuk nilai int

double pi = 3.141592653589793; // doubles are higher precision

double large = 1.3e100; // using the exponent notation
```

2. Data Type ~ Conversion

```java
// konversi int to double
int i=200;
double d=i;
System.out.println(d); // 200.0

// konversi double to int
double di=3.14;
int id=(int)di;
System.out.println(id); //3

// int to long
int il=200;
long l=il;
System.out.println(il); //200

// long to int
long lt=500;
int it=(int)lt;
System.out.println(it); //500
```

```java
//char to int
char c='a';
char c2='1';
int a=c;
int b=c2;
System.out.println(a); //97
System.out.println(b); //49

// int to char
int at=65;
char ct=(char)at;
System.out.println(ct); //A
```

```java
// date to timestamp
Date date = new Date();
Timestamp ts=new Timestamp(date.getTime());
System.out.println(ts); //2019-08-21 10:28:48.45

// timestamp to date
Timestamp tst=new Timestamp(System.currentTimeMillis());
Date dates=new Date(tst.getTime());
System.out.println(dates); //Wed Aug 21 10:28:48 ICT 2019
```



2. Operators (1)

Arithmatic Operator

Operator	Use	Description
+	x + y	Adds x and y
-	x - y	Subtracts y from x
	-x	Arithmetically negates x
*	x * y	Multiplies x by y
/	x / y	Divides x by y
%	x % y	Computes the remainder of dividing x by y

Shortcut Operator

Operator	Use	Description
++	x++	y = x++; is the same as y = x; x = x + 1;
	++x	y = ++x; is the same as x = x + 1; y = x;
--	x--	y = x--; is the same as y = x; x = x - 1;
	--x	y = --x; is the same as x = x - 1; y = x;

2. Operator (2)

Relational Operator

Operator	Use	Description
>	x > y	x is greater than y
>=	x >= y	x is greater than or equal to y
<	x < y	x is less than y
<=	x <= y	x is less than or equal to y
==	x == y	x is equal to y
!=	x != y	x is not equal to y

Boolean Operator

Operator	Use	Evaluates to true if
&&	x && y	Both x and y are true
||	x || y	Either x or y are true
!	!x	x is not true

2. Operator (3)

Assignment Operator

Operator	Use	Shortcut for
=	x = y	x = y
+=	x += y	x = x + y
-=	x -= y	x = x - y
*=	x *= y	x = x * y
/=	x /= y	x = x / y
%=	x %= y	x = x % y
&=	x &= y	x = x & y (also works for boolean values)
|=	x |= y	x = x | y (also works for boolean values)
^=	x ^= y	x = x ^ y (also works for boolean values)
>>=	x >>= y	x = x >> y
>>>=	x >>>= y	x = x >>> y
<<=	x <<= y	x = x << y

2. Operator (4)

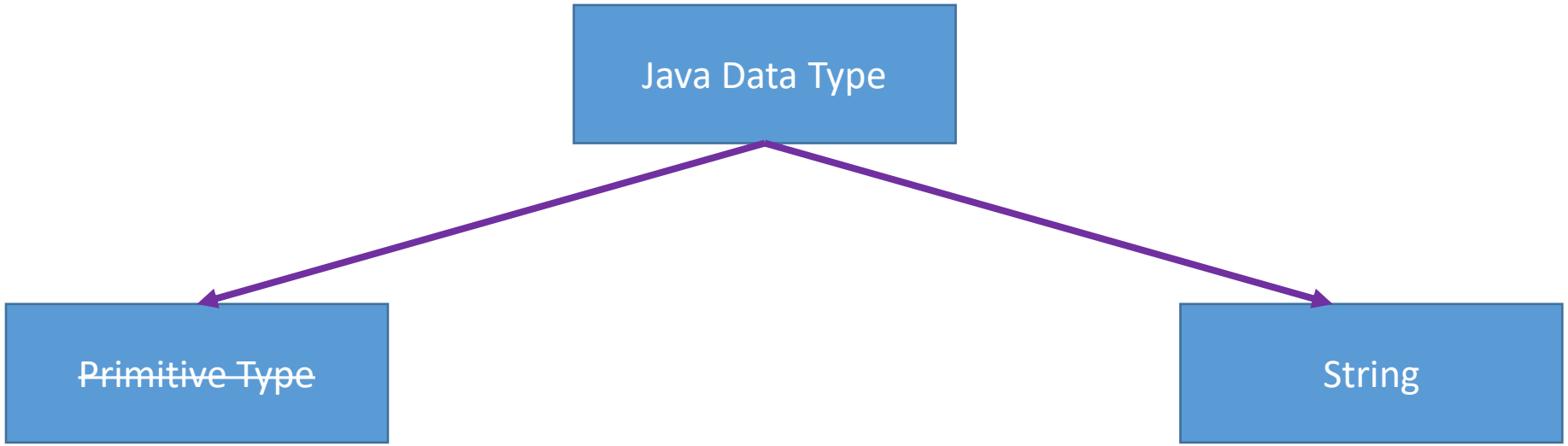
0thers Operator

Operator	Use	Description
()	(x + y) * z	Require operator precedence
?:	z = b ? x : y	Equivalent to: if (b) { z = x; } else { z = y; }
[]	array[0]	Access array element
.	str.length()	Access object method or field
(*type*)	int x = (int) 1.2;	Cast from one type to another
new	d = new Date();	Create a new object
instanceof	o instanceof String	Check for object type, returning boolean

2. Operator (5)

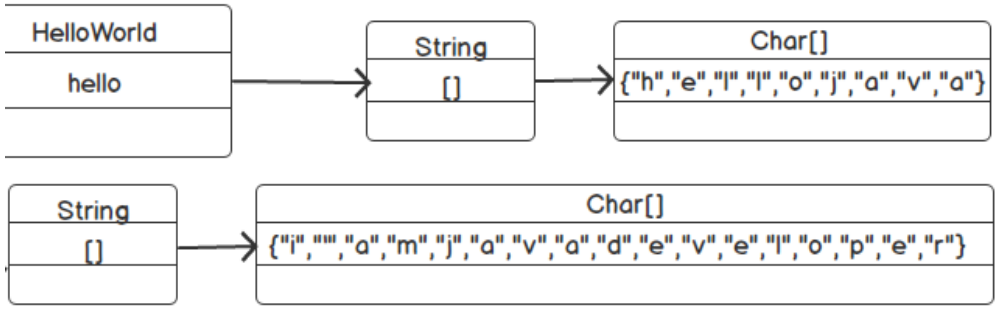
```java
package com.xsis.fundamental.day01.conditions;

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

        boolean isEven = (12 % 2 ==0) ? true :false;
        System.out.println(isEven);

    }

}
```

3. Java Data Type



3. Data Type ~ String

- String adalah object, bukan primitive type.
- String tidak bisa diperlakukan seperti array char
- String ditulis diawali dan ditutup dengan karakter " "
- String bersifat immutable :
    - artinya sekali di assign dengan sebuah value, valuenya tidak bisa diubah.
    - tapi string masih bisa di modif value nya dengan meng-create string object baru
- String memiliki beberapa method :
    - charAt,length,replace, substring, indexOf, equals, trim, split, toUpperCase, endsWith

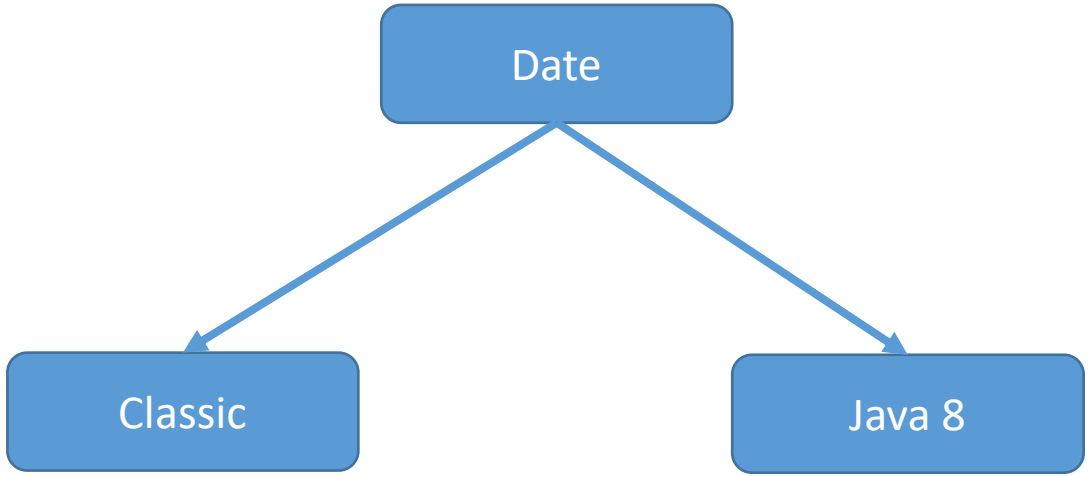


3. Data Type ~ Strings Example

```
String first = "xsis";
String last = "academy";

String name = first + " " + last;   // "xsis academy"

System.out.println(name.toUpperCase());   // XSIS ACADEMY
System.out.println(name.toLowerCase());  // xsis academy
System.out.println(name.length());      //  12
System.out.println(name.contains("xsis")); //boolean: true
System.out.println(name.lastIndexOf("academy")); // lastindes posisi : 5
System.out.println(name.replace("xsis","Xsis Winner")); // Xsis Winner academy
System.out.println(name.replaceAll("a","i")); //xsis icidemy
System.out.println(name.substring(0,4)); // xsis
System.out.println(name.indexOf("a")); // position index : 5
System.out.println(name.startsWith("x")); // true
System.out.println(name.endsWith("y")); // true

char chr = name.charAt(6);
System.out.println("char: "+chr); // char:c

int count =8;
String msg = "There are "+count+" ducks."; // There are 8 ducks

String s1 = "xsis";
String s2 = "xsis";
System.out.println(s1.equals(s2)); // true

String s3 = "XSIS";
String s4 = "xsis";
System.out.println(s3.equalsIgnoreCase(s4)); // true
```

```
String joinString1=String.join("-","xsis","academy","winners");
System.out.println(joinString1); //xsis-academy-winners

String s5="xsis academy winner java developer";
String[] words=s5.split("\\s");//splits the string based on whitespace
//using java foreach loop to print elements of string array
for(String w:words){
    System.out.println(w);
    //xsis
    //academy
    //winner
    //java
    //developer
}

// valueOf, konversi different data type to string
float f  = 10.05f;
double d = 10.02;
String s6 = String.valueOf(f);
String s7 = String.valueOf(d);
System.out.println(s6); //10.05

// trim, menghapus spasi
String s8="  hello xsis   ";
System.out.println(s8+"academy");//without trim()
System.out.println(s8.trim()+"academy");//with trim()
```

4. DateTime




```
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;
```

```
import java.time.LocalDate;
import java.time.format.DateTimeFormatter;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
```

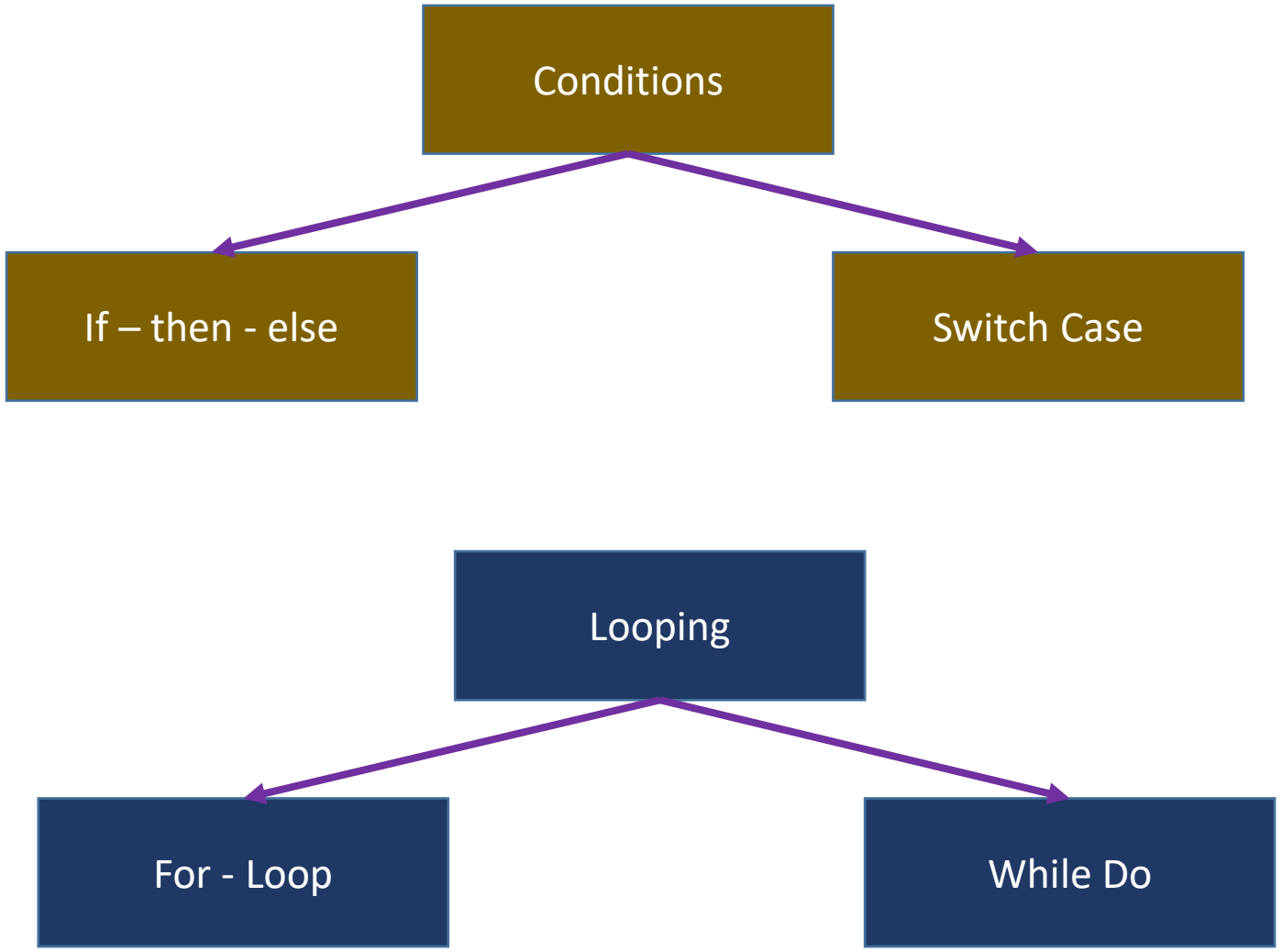
4. Java Date Classic

```java
package com.xsis.fundamental.day01.datetime;

import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;

public class DateTimes {

    public static void main(String[] args) {
        //classic
        Date currentDate  = new Date();
        System.out.println(currentDate); //Wed Aug 21 10:34:46 ICT 2019

        SimpleDateFormat formatter = new SimpleDateFormat("dd/MM/yyyy");
        String strDate= formatter.format(currentDate);
        System.out.println(strDate); //21/08/2019

        String dateToStr = DateFormat.getInstance().format(currentDate);
        System.out.println(dateToStr); //Date Format using getInstance(): 8/21/19 10:34 AM

        dateToStr = DateFormat.getDateInstance().format(currentDate);
        System.out.println(dateToStr); //Date Format using getDateInstance(): Aug 21, 2019

        dateToStr = DateFormat.getTimeInstance().format(currentDate);
        System.out.println(dateToStr);//Date Format using getTimeInstance(): 10:34:46 AM

        dateToStr = DateFormat.getDateTimeInstance().format(currentDate);
        System.out.println(dateToStr);//Date Format using getDateTimeInstance(): Aug 21, 2019 10:34:46 AM

        dateToStr = DateFormat.getTimeInstance(DateFormat.SHORT).format(currentDate);
        System.out.println("DateFormat.SHORT: "+dateToStr);//DateFormat.SHORT: 10:34 AM

        dateToStr = DateFormat.getTimeInstance(DateFormat.MEDIUM).format(currentDate);
        System.out.println("DateFormat.MEDIUM: "+dateToStr);//DateFormat.MEDIUM: 10:34:46 AM

        dateToStr = DateFormat.getTimeInstance(DateFormat.LONG).format(currentDate);
        System.out.println("DateFormat.LONG: "+dateToStr);//DateFormat.LONG: 10:34:46 AM ICT
    }
}
```

4. Java 8 DateTime

```java
package com.xsis.fundamental.day01;

import java.time.LocalDate;
import java.time.format.DateTimeFormatter;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;

public class DateTimeJava8 {
    public static void main(String[] args) {
        LocalDate date = LocalDate.now();
        LocalDate yesterday = date.minusDays(1);
        LocalDate tomorrow = yesterday.plusDays(2);
        System.out.println("Today date: "+date); //Today date: 2019-08-21
        System.out.println("Yesterday date: "+yesterday); //Yesterday date: 2019-08-20
        System.out.println("Tommorow date: "+tomorrow); //Tommorow date: 2019-08-22

        LocalDateTime now = LocalDateTime.now();
        System.out.println("Before Formatting: " + now); //Before Formatting: 2019-08-21T10:42:05.760
        DateTimeFormatter format = DateTimeFormatter.ofPattern("dd-MM-yyyy HH:mm:ss");
        String formatDateTime = now.format(format);
        System.out.println("After Formatting: " + formatDateTime); //After Formatting: 21-08-2019 10:42:05

    }

}
```

5. Condition


Conditions
If – then - else
Switch Case
Looping
For - Loop
While Do

5. Conditions If-Then-Else

Statement

```
if (boolean-expression) {
    // Run if BE is true
} else if (boolean-expression2) {
    // Run if BE is false and BE2 is true
} else {
    // Run if both BE and BE2 are false
}
```

```
package com.xsis.fundamental.day01.conditions;

import java.util.Scanner;

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

        Scanner scan = new Scanner(System.in);
        System.out.println("Input hasil test :  ");

        int score = scan.nextInt();
        char grade;

        if (score >= 90) {
            grade = 'A';
        } else if (score >= 80) {
            grade = 'B';
        } else if (score >= 70) {
            grade = 'C';
        } else if (score >= 60) {
            grade = 'D';
        } else {
            grade = 'F';
        }

        System.out.println("Grade = " + grade);

    }
}
```

5. Conditions Switch

Statement

```
switch(expression) {
    case const1:
        /* do X */
        break;
    case const2:
        /* do Y */
        break;
    default:
        /* do something else */
}
```

Keyword ***break*** digunakan untuk keluar dari statement

Keyword ***continue*** digunakan untuk lanjut ke statement berikutnya.

```
for (int i = 0; i < 10; i++){
    if ( i % 2 != 0){
        continue;
    }
    System.out.println("idx : "+i);
}
```

```
package com.xsis.fundamental.day01.conditions;

import java.util.Scanner;

public class MonthSwitch {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.println("Input bulan dalam angka :  ");

        int month = scan.nextInt();

        switch (month){
            case 1:  System.out.println("January"); break;
            case 2:  System.out.println("February"); break;
            case 3:  System.out.println("March"); break;
            case 4:  System.out.println("April"); break;
            case 5:  System.out.println("May"); break;
            case 6:  System.out.println("June"); break;
            case 7:  System.out.println("July"); break;
            case 8:  System.out.println("August"); break;
            case 9:  System.out.println("September"); break;
            case 10: System.out.println("October"); break;
            case 11: System.out.println("November"); break;
            case 12: System.out.println("December"); break;
            default: System.out.println("Invalid month: " + month);
        }
    }
}
```

5. For Loop

Statement

```
for (init; condition; update) {
        // Do something
}
```

```
package com.xsis.fundamental.day01.conditions;

public class ForLoop {
    public static void main(String[] args) {
        for (int i=0; i < 10; i++){
            System.out.println(i);
        }
    }
}
```

Output

```
/* 0
 1
 2
 3
 4
 5
 6
 7
 8
 9*/
```

5. While do

Statement

```
while (boolean-expression) {
        // Do something repeatedly
        // Update condition
        }
```

```
package com.xsis.fundamental.day01.conditions;

public class WhileDo {
    public static void main(String[] args) {
        int i = 0;
        while (i < 10) {
            System.out.println(i);
            i += 1;
        }
    }
}
```

Output

```
/* 0
1
2
3
4
5
6
7
8
9*/
```

5. Do While

Statement

```
do {
        // Do something at least once
} while (boolean-expression);
```

```
package com.xsis.fundamental.day01.conditions;

public class DoWhile {
    public static void main(String[] args) {
        int i =0;
        do {
            System.out.println(i);
            i += 1;
        } while (i<10);
    }
}
```

Output

```
/* 0
 1
 2
 3
 4
 5
 6
 7
 8
 9*/
```

6. Arrays

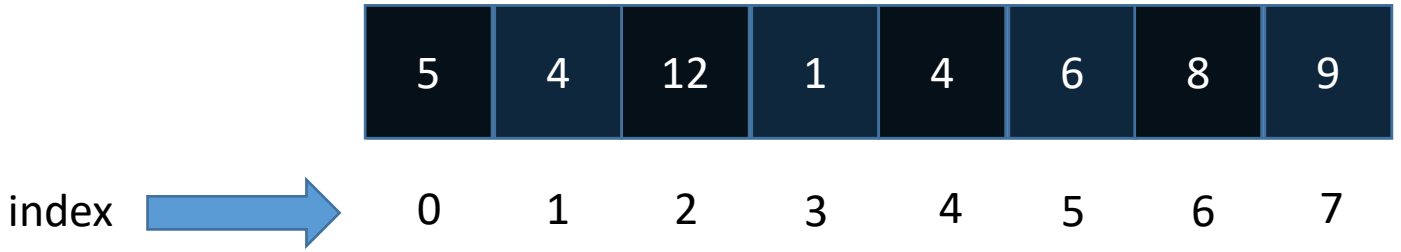
An *array* is a simple data structure to hold a series of data elements of the same type

Array Declaration :

```
int[] cars;

int bikes[];

double values[][]; // single or multi-dimensional array
```

Index element array di mulai dari 0

```
int[] values = { 5, 4, 12, 1, 4, 6, 8, 9 }; // 7 element
```



6. Array Initializing

Initialiazing

```
// initialiasi arrays
int[] a = {1,2,3,4,5,6,7};
String[] s = {"hello", "xsis", "academy"};
Double[] d = {1.2,2.00,4.5};
int[][] matrix ={{1,2,3,4},{4,1,2,3}};

// initialisasi array with int by each element
int[] ar= new int[3];
ar[0]=1;
ar[1]=2;
ar[2]=3;

// initialisasi array with string by each element
String[] ars= new String[3];
ars[0]="Hello";
ars[1]="Java";
ars[2]="Developer";
```

Access

```
// access arrays
System.out.println(a[1]); //2
System.out.println(a[5]);//6
System.out.println(s[0]); //hello
System.out.println(d[1]); //2.0

// access multidimensional array
System.out.println(matrix[0][0]); //1
System.out.println(matrix[0][1]); //2
```

```
for (int i = 0; i < s.length; i++){
    System.out.println("index:"+i+" : "+s[i]);
/*        index:0 : hello
index:1 : xsis
index:2 : academy*/
}
```