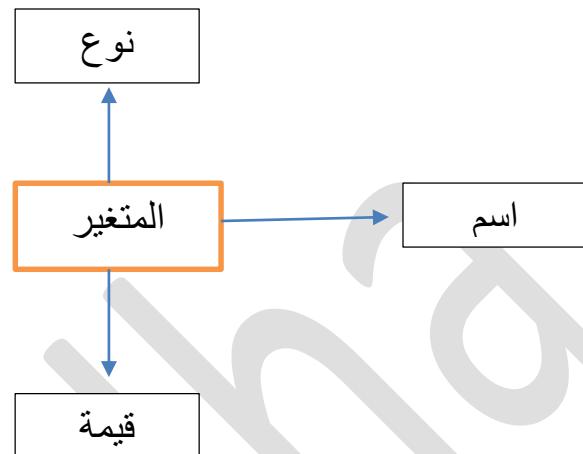


- ❖ المتغيرات اساس البرمجة.
- ❖ المتغيرات على أنواع.
- ❖ لا وجود للبرمجة بدون متغيرات.
- ❖ كيفية تعریف المتغيرات.



```

int A;
int a;
int MyVariable; ≠ int myVariable;

```

Example:

```

myVariable : string;
myVariable = “Hello”;

```

- ❖ لا يمكن استخدام المتغيرات دون تعریفها.
- ❖ لذلك أول شيء نتعلم هو كيفية تعریف المتغيرات.

Declaring Variable in Java

```
TypeOfVariable NameOfVariable ;
```

- ❖ في لغة جافا كل الأوامر والابعادات يجب أن تنتهي بـ (;).

Types of Variable in Java

❖ الأعداد الصحيحة (integer):

byte

► يستخدم هذا النوع من البيانات لحفظ أو تخزين الأعداد الصغيرة والتي لا تحتوي على فاصلة (عدد صحيح) والتي تتراوح بين:

From – 128 To 127

short (2 byte)

► يستخدم هذا النوع من البيانات لحفظ أو تخزين الأعداد الوسط والتي لا تحتوي على فاصلة (عدد صحيح) والتي تتراوح بين:

From – 32768 To + 32767

int (4 byte)

► هذا النوع من البيانات هو الأكثر استخداماً في البرامج، يستخدم هذا النوع من البيانات لحفظ أو تخزين الأعداد الكبيرة والتي لا تحتوي على فاصلة (عدد صحيح) والتي تتراوح بين:

From – 2147483648 To + 2147483647

long (8 byte)

► يستخدم هذا النوع من البيانات لحفظ أو تخزين الأعداد الكبيرة جداً والتي لا تحتوي على فاصلة (عدد صحيح) والتي تتراوح بين:

From – 9223372036854775808 To + 9223372036854775807

❖ لتعريف أي عدد صحيح (لا يحتوي على فاصلة) نستخدم هذه الأنواع الأربع أعلاه وحسب حجم العدد.

byte, int, short, long

float (4 byte)

- ❖ يستخدم هذا النوع من البيانات لحفظ أو تخزين الأعداد العشرية (الكسرية) الكبيرة والتي تحتوي على فاصلة (ليس عدد صحيح).

double (8 byte)

- ❖ يستخدم هذا النوع من البيانات لحفظ أو تخزين الأعداد العشرية (الكسرية) الكبيرة جداً والتي تحتوي على فاصلة (ليس عدد صحيح).

boolean (one bit)

true, false

string

- ❖ يستخدم هذا النوع من البيانات لحفظ أو تخزين النصوص الكتابية.

char (2 byte)

- ❖ يستخدم هذا النوع من البيانات لحفظ أو تخزين الرموز (حروف + رموز) ويستخدم قليل في البرامج.

Java Type Casting

Type casting is when you assign a value of one primitive data type to another type.

In Java, there are two types of casting:

- Widening Casting (automatically) converting a smaller type to a larger type size.
byte -> short -> char -> int -> long -> float -> double
- Narrowing Casting (manually)- converting a larger type to a smaller size type.
double -> float -> long -> int -> char -> short -> byte

Widening Casting

Widening casting is done automatically when passing a smaller size type to a larger size type:

Example:

```
public class Main
{
    public static void main(String[] args)
    {
        int myInt = 9;
        double myDouble = myInt;
        // Automatic casting: int to double
        System.out.println(myInt);          // Outputs 9
        System.out.println(myDouble);        // Outputs 9.0
    }
}
```

Narrowing Casting

Narrowing casting must be done manually by placing the type in parentheses in front of the value:

Example:

```
public class Main {  
    public static void main(String[] args)  
    {  
        double myDouble = 9.78d;  
        int myInt = (int) myDouble;      // Manual casting: double to int  
        System.out.println(myDouble);   // Outputs 9.78  
        System.out.println(myInt);      // Outputs 9  
    }  
}
```

Java Operators

➤ Arithmetic operators

(`--, ++, %, /, *, -, +`)

➤ Assignment operators

(`=%, /=, *=, -=, +=, =`)

➤ Comparison operators

(`=!, ==, =<, <, =>, >`)

➤ Logical operators

(`!, ||, &&`)

Arithmetic Operators

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	$--x$

Java Assignment Operators

Operator	Example	Same As
=	$x = 5$	$x = 5$
+=	$x += 3$	$x = x + 3$
-=	$x -= 3$	$x = x - 3$
*=	$x *= 3$	$x = x * 3$
/=	$x /= 3$	$x = x / 3$
%=	$x \%= 3$	$x = x \% 3$

Java Comparison Operators

Operator	Name	Example
<code>==</code>	Equal to	<code>x == y</code>
<code>!=</code>	Not equal	<code>x != y</code>
<code>></code>	Greater than	<code>x > y</code>
<code><</code>	Less than	<code>x < y</code>
<code>>=</code>	Greater than or equal to	<code>x >= y</code>
<code><=</code>	Less than or equal to	<code>x <= y</code>

Java Logical Operators

Operator	Name	Description	Example
<code>&&</code>	Logical and	Returns true if both statements are true	<code>x < 5 && x < 10</code>
<code> </code>	Logical or	Returns true if one of the statements is true	<code>x < 5 x < 4</code>
<code>!</code>	Logical not	Reverse the result, returns false if the result is true	<code>!(x < 5 && x < 10)</code>

Escape Characters in Java Program

Escape Characters	Description
\t	It is used to insert a tab in the text at this point.
'\	It is used to insert a single quote character in the text at this point.
"\	It is used to insert a double quote character in the text at this point.
\r	It is used to insert a carriage return in the text at this point.
\\\	It is used to insert a backslash character in the text at this point.
\n	It is used to insert a new line in the text at this point.

Java output / Print

❖ Print Text

You can add as many `println()` methods as you want. Note that it will add a new line for each method:

Example:

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

```
System.out.println("I am learning Java.");
```

```
System.out.println("It is awesome!");
```

❖ Double Quotes

When you are working with text, it must be wrapped inside double quotations marks " ".

If you forget the double quotes, an error occurs:

Example:

```
System.out.println("This sentence will work!");
```

```
System.out.println(This sentence will produce an error); X
```

❖ The Print() Method

There is also a print() method, which is similar to println().

The only difference is that it does not insert a new line at the end of the output:

Example:

```
System.out.print("Hello World! ");
```

```
System.out.print("I will print on the same line.");
```

❖ Print Numbers

You can also use the println() method to print numbers.

However, unlike text, we don't put numbers inside double quotes:

Example:

```
System.out.println(3);
```

```
System.out.println(358);
```

```
System.out.println(50000);
```

You can also perform mathematical calculations inside the println() method:

```
System.out.println(3 + 3);
```

```
System.out.println(2 * 5);
```

❖ Concatenation of Strings in Java

We can use + operator to concatenate text to another text, or to concatenate text and a numeric value in print sentences.

Example:

```
int A = 20 ; float B = 2.4 ;
```

```
String S1 = " Ali " ;
```

```
String S2 = " 40 " ;
```

```
System.out.print ("A = " + A + "\t" + "B = " + B);
```

```
System.out.print ("A = " + A + " " + "B = " + B);
```

```
System.out.print (S1 + " " + S2);
```

Homework

Use print() with code to get the following output

```
Int A=20;
```

```
Int B=30;
```

Output: The result is: 50

Example # 1:

Write a program in Java to print “Hellow Java”?

The code is:

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

Example # 2:

Write a program in Java to print “Happy New Year 2026”?

The code is:

```
public class simple {  
    public static void main(String[ ] args) {  
        System.out.println("Happy New Year 2026");  
    }  
}
```

Example # 3:

Write a program in Java to calculate the value of $a = 3 + 4$?

The code is:

```
public class summation {  
    public static void main(String[ ] args) {  
        int a = 3 + 4;  
        System.out.println(a);  
    }  
}
```

Example # 4:

Write a program in Java to calculate the value of $a = 3 + 4$?

The code is:

```
public class summation {  
    public static void main(String[ ] args) {  
        int a = 3;  
        int b = 4;  
        System.out.println(a + b);  
    }  
}
```

Example # 5:

Write a program in Java to calculate the value of $y = a + b$?

The code is:

```
public class summation {  
    public static void main(String[ ] args) {  
        int a = 3;  
        int b = 4;  
        y = a + b;  
        System.out.println(y);  
    }  
}
```

Example # 6:

Write a program to read two numbers, then find the summation between them?

The code is:

```
public class summation {  
    public static void main(String[ ] args) {  
        int x = 10;  
        int y = 20;  
        z = a + b;  
        System.out.println(y);  
    }  
}
```

Example # 7:

Write a program to calculate the value of y from the following equation:

$$Y = X + 5 ?$$

The code is:

```
public class summation {  
    public static void main(String[ ] args) {  
        int x = 10;  
        int y;  
        y = x + 5;  
        System.out.println(y);  
    }  
}
```

Thank You

د. محمد رضا الكعبي

Mobile: 07801582655

E-mail: mridha.hamoodi@uokufa.edu.iq

"الحياة تعلمك الحب،

والتجارب تعلمك من تحب،

والمواقف تعلمك من يحبك "