

البرمجة بلغة جافا

الفصل الأول

مقدمة:

جافا هي لغة برمجة عالية المستوى وموجهة نحوي الكائنات وأمنة، تم إنشاؤها عام ١٩٩٢. من قبل شركة Oracle ولها تطبيقات كثيرة منها:

- تطبيقات الجوال .
 - تطبيقات سطح المكتب.
 - تطبيقات ويب .
 - خوادم الويب وخوادم التطبيقات .
 - الألعاب.
 - اتصال قاعدة البيانات الخ.....
- وتستخدم لغة جافا لأسباب كثيرة منها:
١. **تعمل على منصات مختلفة** (Windows, Mac, Linux, Raspberry Pi, etc.).
 ٢. واحدة من أكثر لغات البرمجة شعبية في العالم.
 ٣. سهولة التعلم والاستخدام.
 ٤. مفتوحة المصدر.
 ٥. قريبة من لغة ++c و c# فمن السهل على المبرمجين التبديل إلى جافا . الخ.....

ملاحظة: يمكن كتابة كود جافا في برنامج IDE **Netbeans**.

(Integrated Development Environment) بيئة تطوير متكاملة.

أول برنامج بلغة جافا:

```
//First program
public class Firstjava {
    public static void main(String args[]) {
        System.out.println("welcome to java");
    }
}
```

شرح البرنامج:

السطر الأول تعليق وهو لا ينفذ.

السطر الثاني تعريف الـ class للإعلان عن صف وكلمة public تعني يمكن أن نستخدم ما داخل الصف في أي class آخر، واسم الـ class هو Firstjava ويجب أن يبدأ بحرف كبير. وهذا يعني أن الحروف الكبيرة تختلف عن الحروف الصغيرة بالنسبة للمترجم.

السطر الثالث الدالة الرئيسية main وهي لا تعيد قيمة فهي void وعامة public. و static هي أنه ليس هناك حاجة لإنشاء كائن لاستدعاء التابع الثابت (main موجودة في كل برنامج java).

السطر الرابع هو أمر الطباعة في console ويطبع الجملة الموجودة بين علامتي التنصيص وفي مثالنا (welcome to java). وكل أمر بلغة جافا ينتهي بفاصلة منقوطة (;).

ملاحظة هامة:

يجب أن يتطابق اسم الملف مع اسم الـ class. عند حفظ الملف نكتب اسم الملف الذي نختاره ويكون الامتداد هو (.java) والـ class يأخذ هذا الاسم تلقائياً.

الترجمة والتنفيذ:

إذا لا يوجد أخطاء في البرنامج فعند التنفيذ المفسر يقوم بإنشاء وحفظ ملف جديد يسمى Firstjava.class يحتوي على **Byte Code**، وهي الصورة التنفيذية للبرنامج.

التعليق في جافا:

يمكن استخدام التعليقات لشرح البرنامج وتعطي فكرة عن وظيفة كل أمر فيه، وتساعد التعليقات على قراءة البرنامج وفهمه، والمترجم يقوم بتجاهل جميع أسطر التعليقات أثناء عملية الترجمة. ويوجد طريقتين لكتابة التعليق في لغة الجافا :

الطريقة الأولى التعليق على سطر واحد ويبدأ بالرمز //
الطريقة الثانية التعليق على أكثر من سطر ويكون كما يلي:

```
/*.....  
.....التعليق.....  
.....*/
```

أنواع المؤثرات في لغة جافا:

المؤثر: هو عبارة عن رمز رياضي يقوم بإجراء عملية معينة على معاملاته والمعامل هو القيمة التي يجري عليها المؤثر العملية. ويوجد في لغة جافا عدد من المؤثرات وهي:

Arithmetic Operators

المؤثرات الحسابية:

وتستخدم لأداء العمليات الحسابية المشتركة

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

Comparison Operators

مؤثرات المقارنة:

تستخدم لمقارنة قيمتين ويكون الناتج إما True أو False.

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

Logical Operators

المؤثرات المنطقية:

تستخدم لربط أكثر من تعبير منطقي مع بعضها البعض.

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

أولية التنفيذ من اليسار إلى اليمين NOT → AND → OR

والجدول التالي يوضح الصواب والخطأ لهذه المؤثرات:

X	Y	$X \ \&\& \ Y$	$X \ \ Y$	$!X$
True	True	True	True	False
True	False	False	True	False
False	True	False	True	True
False	False	False	False	True

Examples

```
int x = 5;      System.out.println(x > 3 && x < 10);      //out true
int x = 5;      System.out.println(x > 3 || x < 4);        //out true
int x = 5;      System.out.println(!(x > 3 && x < 10));    //out false
```

compound assignation operators

مؤثرات التعيين المركبة:

يوجد في لغة جافا عددا من عمليات الإسناد وهي:

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

Examples

```
int x = 5;      x += 3;      System.out.println(x);      //out 8
int x = 5;      x -= 3;      System.out.println(x);      //out 2
int x = 5;      x *= 3;      System.out.println(x);      //out 15
double x = 5;   x /= 3;      System.out.println(x);      //out 1.6666
int x = 5;      x %= 3;      System.out.println(x);      //out 2
int x = 5;      x %= 8;      System.out.println(x);      //out 5
```

Increment and Decrement Operators

مؤثرات الزيادة والنقصان:

Operator	Example (i = 1)
++var	int j = ++i; // j is 2, // i is 2
var++	int j = i++; // j is 1, // i is 2
--var	int j = --i; // j is 0, // i is 0
var--	int j = i--; // j is 1, // i is 0

```
int x = 6;      System.out.println(x);      //out 6
++x;
System.out.println(x);                      //out 7
System.out.println(++x);                    //out 8
System.out.println(x++);                    //out 8    x=9 بالذاكرة تصبح
System.out.println(x);                      //out 9
System.out.println(--x);                     //out 8
System.out.println(x);                      //out 8
System.out.println(x--);                     //out 8    x=7 بالذاكرة تصبح
System.out.println(x);                      //out 7
```

أنواع البيانات البدائية:

تحدد أنواع البيانات البدائية حجم ونوع القيم المتغيرة، وهناك ثمانية أنواع في لغة جافا وهي:

Data Type	Size	Description
Byte	1 byte	Stores whole numbers from -128 to 127
Short	2 bytes	Stores whole numbers from -32,768 to 32,767
Int	4 bytes	Stores whole numbers from -2,147,483,648 to 2,147,483,647
Long	8 bytes	Stores whole numbers from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
Float	4 bytes	Stores fractional numbers. Sufficient for storing 6 to 7 decimal digits
Double	8 bytes	Stores fractional numbers. Sufficient for storing 15 decimal digits
Boolean	1 bit	Stores true or false values
Char	2 bytes	Stores a single character/letter or ASCII values

أنواع البيانات غير البدائية:

ومنها **String, Arrays and Classes**

Java Identifiers

شروط تسمية المتغيرات والصفوف: (الاسم المعرفي)

- يمكن أن تكون من الحروف الانكليزية (A ... Z & a z) والأرقام من (0.....9)
- لا يمكن أن يبدأ الاسم برقم.
- أن لا يكون الاسم إحدى الكلمات المحجوزة.
- الأسماء حساسة لحالة الأحرف فمثلا mark تختلف عن Mark.
- لا يحتوي على مسافات فارغة.

Declaring Variables

الإعلان عن المتغيرات:
المتغيرات هي أسماء تحجز مواقع في الذاكرة لحفظ البيانات داخلها. للتصريح عن المتغير يجب تحديد النوع وإسناد قيمة له كما يلي:
الصيغة العامة:

Type variable = value;

أمثلة:

```
int x;  
    x = 10;  
double y;  
int a , b , c;  
int n1=5 , n2=2;  
char c = 'D';  
String name = "Hassan";
```

Final Variables

تعريف الثوابت:

هي مواقع لحفظ البيانات كالمتغيرات لكن المتغير يبدل قيمته على عكس الثابت الذي يثبت على قيمته الأساسية. للتصريح عن الثوابت نستخدم الأمر Final كما يلي:

```
final int x = 5;  
final float PI = 3.14;
```

Display Variables

طباعة المتغيرات:

نستخدم الأمر `println()` للطباعة، وندمج نص ومتغير نضع (+).

Example

```
String name = "Hassan";  
System.out.println("Hello " + name);
```

//out Hello Hassan

Example

```
String firstName = " Hassan ";  
String lastName = "Ali";  
String fullName = firstName + lastName;  
System.out.println(fullName);
```

//out Hassan Ali

Example

```
int x = 5;  
int y = 6;  
System.out.println(x + y);
```

// Print the value of x + y (output 11)

Character Escape

أحرف الهروب:

هي عبارة عن أحرف تستخدم مع أمر الطباعة

Character Escape	Name
\b	Backspace
\t	Tab
\n	Linefeed
\r	Carriage Return
\\	Backslash
\'	Single Quote
\"	Double Quotes

أولوية تنفيذ المؤثرات الحسابية:

الاتجاه	العملية	رمز العملية
Left to Right	الأقواس	() []
Right to left	النفي ++ -- !	++ -- !
Left to Right	ضرب - قسمة - باقي قسمة	* / %
Left to right	الجمع - الطرح	+ -
Left to right	عمليات المقارنة	< <= > >=
Left to right	الأسناد - لا يساوي	== !=
Left to right	AND	&&
Left to right	OR	
Right to left	المؤثر الشرطي	? :
Right to left	عمليات الاسناد	= += -= *= /= %=

جدول الكلمات المحجوزة:

abstract	continue	Finally	interface	Public	throw
boolean	default	Float	Long	Return	throws
break	do	For	native	Short	transient
byte	double	If	New	Static	true
case	else	Implements	Null	Super	try
catch	extends	Import	package	Switch	void
char	false	Instanceof	private	synchronized	volatile
class	final	Int	protected	This	While

Java Type Casting

طرق التحويل بين أنواع البيانات

١ - تلقائياً: تحويل من نوع أصغر إلى حجم نوع أكبر (توسيع) Widening Casting (automatically)

byte -> short -> char -> int -> long -> float -> double

Example

```
public class TypeCasting {
    public static void main(String args[]) {
        int x_Int = 5;
        double y_Double = x_Int;
        System.out.println(x_Int);
        System.out.println(y_Double);
    } }
```

// Automatic casting: int to double
// Outputs 5
// Outputs 5.0

٢- يدويا: تحويل من نوع أكبر إلى حجم نوع أصغر (تضييق) Narrowing Casting (manually)

double -> float -> long -> int -> char -> short -> byte

Example

```
public class TypeCasting {
    public static void main(String args[]) {
        double y_Double = 5.67;
        int x_Int = (int) y_Double;           // Manual casting: double to int
        System.out.println(y_Double);        // Outputs 5.67
        System.out.println(x_Int);           // Outputs 5
    }
}
```

Casting between char and Numeric Types

التحويل بين char وأنواع الأرقام:

Examples

```
char ch = (char) 65;           // 65 is assigned to ch
System.out.println(ch);        // ch is character A
int i = (int) 'A';             // the character A is assigned to i
System.out.println(i);         // i is 65
```

Reading Input from the Console

القراءة وإدخال القيم من console:

في لغة جافا نستخدم الأمر `System.out.println` للطباعة أما بالنسبة للقراءة وإدخال القيم والبيانات نستخدم الأمر `System.in` وجافا لا تدعم القراءة المباشرة، لكن يجب أن نستخدم `Scanner` class كما يلي:

```
Scanner input = new Scanner(System.in);
```

الجملة `new Scanner(System.in)` تنشئ كائن من نوع `Scanner`، والجملة `Scanner input` تعلن أن المتغير المدخل هو من نوع `Scanner`.

يمكن استدعاء `the methods` لقراءة أنواع مختلفة من المدخلات كما في الجدول التالي:

Methods for Scanner Objects

Method	Description
<code>nextByte()</code>	reads an integer of the byte type.
<code>nextShort()</code>	reads an integer of the short type.
<code>nextInt()</code>	reads an integer of the int type.
<code>nextLong()</code>	reads an integer of the long type.
<code>nextFloat()</code>	reads a number of the float type.
<code>nextDouble()</code>	reads a number of the double type.
<code>next()</code>	reads a string that ends before a whitespace character.
<code>nextLine()</code>	reads a line of text (i.e., a string ending with the Enter key pressed).

مثال:

اكتب برنامجا يطلب من المستخدم إدخال نصف قطر دائرة ثم يحسب ويطبع مساحة هذه الدائرة.

الحل: / console/

```
import java.util.Scanner;
public class InputConsole {
    public static void main(String args[]) {
        Scanner input = new Scanner(System.in);           // Create a Scanner object
        System.out.print("Enter a number for radius: ");
        double radius = input.nextDouble();
        double area = radius * radius * 3.14159;
        System.out.println("area="+area);
    }
}
```

Output

Enter a number for radius: 2

area=12.56636

The Scanner class is in the java.util package. It is imported in line 1.

مثال:

اكتب برنامجا يطلب من المستخدم إدخال ثلاث أعداد ثم يحسب ويطبع المتوسط الحسابي لهذه الأعداد.

الحل: / console/

```
import java.util.Scanner;
public class InputConsoleAverage {
    public static void main(String args[]) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter three numbers: ");           //println
        double number1 = input.nextDouble();
        double number2 = input.nextDouble();
        double number3 = input.nextDouble();
        double average = (number1 + number2 + number3) / 3;
        System.out.println(" average = " + average);
    }
}
```

Output

Enter three numbers: 3 4 6

// enter input in one line

The average of 4.333333333333333

Enter three numbers: 9

// enter input in multiple lines

8

11

average = 9.333333333333334

Strings في جافا:

يوجد عدة طرق (method) نستخدمها مع السلاسل ومنها:

- يمكن معرفة طول السلسلة باستخدام method **length()**.

Example

```
String txt = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";  
System.out.println("The length of the txt string is: " + txt.length());  
//out The length of the txt string is: 26
```

- يمكن طباعة السلسلة بأحرف كبيرة باستخدام method **toUpperCase()** ولطباعة السلسلة بأحرف صغيرة نستخدم method **toLowerCase()**.

Example

```
String txt = "Hello World";  
System.out.println(txt.toUpperCase()); // out "HELLO WORLD"  
System.out.println(txt.toLowerCase()); // out "hello world"
```

- يمكن معرفة أول موقع للنص المحدد باستخدام method **indexOf()** عند حساب مكان النص يبدأ العد من الصفر ويعد الرموز والفراغات.

Example

```
String txt = "Welcome To java ";  
System.out.println(txt.indexOf("To")); // out 8
```

- دمج سلسلتين يمكن استخدام method **concat ()**

Example

```
String firstName = "Hassan";  
String lastName = "Ali";  
System.out.println(firstName.concat(lastName)); //out HassanAli
```

Java Math

الدوال الرياضية:

تحتوي على العديد من الطرق (method) لأداء المهام الرياضية على الأرقام ومن هذه الطرق:

Math.max(x,y) - ١

يمكن استخدام **Math.max(x,y)** التي تعيد أكبر قيمة بين العددين **x** و **y**.

Example

```
System.out.println(Math.max(5, 10)); //out 10
```

Math.min(x,y) - ٢

يمكن استخدام **Math.min(x,y)** التي تعيد أصغر قيمة بين العددين **x** و **y**.

Example

```
System.out.println(Math.min(5, 10)); //out 5
```

Math.sqrt(x) - ٣

الطريقة Math.sqrt (x) تعيد الجذر التربيعي لـ x .

Example

```
System.out.println(Math.sqrt(64));           //out 8.0
```

Math.abs(x) - ٤

الطريقة Math.abs (x) تعيد القيمة المطلقة (الموجبة) لـ x .

Example

```
System.out.println(Math.abs(-4.7));          //out 4.7
```

Math.random() - ٥

يقوم Math.random () بإرجاع رقم عشوائي بين 0 و 1 .

Example

```
System.out.println(Math.random());           //out 0.6176503626325601
```

Math.pow(x, y) - ٦

تُرجع قيمة x^y .

Example

```
System.out.println (Math.pow(2, 4) );        //out 16
```

java boolean

القيم المنطقية:

في البرمجة ، نحتاج إلى نوع بيانات لا يمكن أن يحتوي إلا على واحدة من قيمتين ، مثل

- YES / NO
- ON / OFF
- TRUE / FALSE

يتم تعريف نوع منطقي باستخدام الكلمة الأساسية **boolean** كما يلي:

Example

```

Boolean b1 = true;
boolean b2 = false;
System.out.println(b1);           // out true
System.out.println(b2);           // out false

```

Boolean Expression

التعبير منطقي:

التعبير المنطقي يُرجع قيمة منطقية صواب أم خطأ. ويمكن استخدام مؤثرات المقارنة كما يلي:

Example

```

int x = 10;
int y = 9;
System.out.println(x > y);        // out true

```

Example

```
int x = 10;  
System.out.println(x == 10);           // out true  
System.out.println(10 == 15);          // out false
```

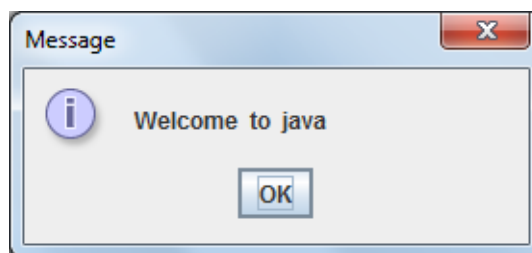
عرض النص في مربع حوار رسالة (GUI):

(GUI) Displaying Text in a Message Dialog Box:

لعرض النص في مربع حوار رسالة، يجب استخدام الطريقة (method) **showMessageDialog** في فئة **JOptionPane** كما يلي:

Example

```
import javax.swing. JOptionPane;           //or import javax.swing.*; استيراد  
public class FirstJava {  
    public static void main(String args[]) {  
        JOptionPane.showMessageDialog(null,"Welcome to java");  
    }  
}
```



شرح البرنامج:

- 1- عن طريق الأمر **import** يتم استدعاء الكائن **JOptionPane** من الحزمة المسماة **javax.swing**. وهذه الحزمة تحتوي على كثير من الكائنات الخاصة بالرسومات والتعامل مع المستخدم من خلال بيئة الرسومات **GUI graphical user interface** التي تسهل إدخال وإخراج البيانات من خلال مربعات حوار.
- 2- السطر الأخير يشير إلى استدعاء الـ **method** المسماة **showMessageDialog** من الكائن المسمى **JOptionPane** وهذا الـ **method** يطلب مدخلين، الأول "null" وهو يحدد المكان الذي يظهر فيه صندوق الحوار، وفي هذه الحالة كتابة "null" فإن صندوق الحوار سوف يظهر في منتصف الشاشة. أما الثاني فهو النص الذي سوف يظهر داخل مربع الحوار كرسالة.

ملاحظة:

إذا قمت باستبدال **JOptionPane** في السطر الرابع بالجملة **javax.swing.JOptionPane**، فلن تحتاج إلى استيراده في السطر الأول **javax.swing.JOptionPane**. فنكتب البرنامج السابق كما يلي:

```
public class FirstJava {  
    public static void main(String args[]) {  
        javax.swing. JOptionPane.showMessageDialog(null,"Welcome to java");  
    }  
}
```

(GUI) Getting Input from Input Dialogs:

القراءة من مربع الحوار

لقراءة القيم من مربع حوار، يجب استدعاء الطريقة `JOptionPane.showInputDialog` (method) عند تنفيذ هذه الطريقة يتم عرض مربع حوار لإدخال القيم. بعد إدخال سلسلة ، انقر فوق موافق لقبول الإدخال.

ملاحظة هامة: يتم إرجاع الإدخال من مربع الحوار كسلسلة .
هناك عدة طرق لاستخدام طريقة `showInputDialog`.
الأول هو استخدام عبارة مثل هذه:

`JOptionPane.showInputDialog(x);`

حيث أن x عبارة عن سلسلة لرسالة المطالبة .
والآخر هو استخدام عبارة مثل ما يلي:

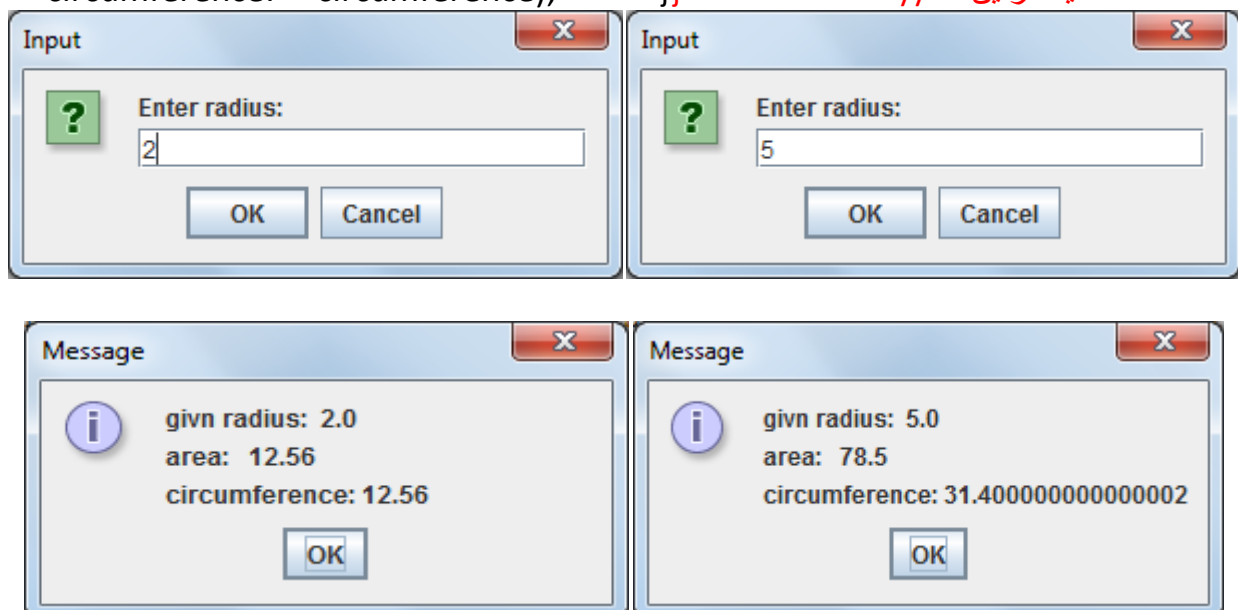
`String string = JOptionPane.showInputDialog(null, x, y, JOptionPane.QUESTION_MESSAGE);`

حيث x هي سلسلة لرسالة المطالبة و y هي سلسلة لعنوان مربع حوار الإدخال ، كما هو موضح في المثال التالي:

Example

computing the area and the circumference of a circle

```
import javax.swing.*;
public class Testconst {
    public static void main(String args[]) {
        final double pi=3.14;
        double radius, area, circumference;
        String radiusStr;
        radiusStr=JOptionPane.showInputDialog(null,"Enter radius:");
        radius=Double.parseDouble( radiusStr);
        area=pi*radius*radius;
        circumference=2.0*pi*radius;
        JOptionPane.showMessageDialog(null,"givn radius: " + radius+"\n"+"area: " +area+
        "\n"+"circumference: "+circumference);    }
    } // التنفيذ مرتين
```



JOptionPane هو كلاس وظيفته إظهار النتائج أو إدخال البيانات ويتعامل مع الحروف فقط في متغير **string** ومن أهم الـ **Methods** :

JOptionPane.showMessageDialog(null, " ");

تستخدم لإظهار المسج الذي بداخل الأقواس

JOptionPane.showInputDialog(" ");

تستخدم لإدخال نص
مثال:

```
int age;
age= JOptionPane.showInputDialog(null,"Enter your age "); //error
string
```

عند التنفيذ يعطينا خطأ لأن عند إدخال رقم من نوع **int** لا يقبل ، يجب التعامل مع الإدخال على أنه **string** فقط ، والحل هو التحويل بين أنواع البيانات. ولذلك يجب أن نضيف متغير من نوع **string** ثم نحوله إلى نوع **int** كما يلي:

```
int age;
String str;
str=JOptionPane.showInputDialog(null,"Enter your age");
age=Integer.parseInt(str);
```

جدول يوضح أهم عمليات التحويل بين أنواع البيانات

Type	Method	Example
Integer	parseInt	Integer.parseInt("25") → 25
Long	parseLong	Long.parseLong("25") → 25L
Float	parseFloat	Float.parseFloat("25.6") → 25.6F
Double	parseDouble	Double.parseDouble("25") → 25.0

Example:

Write a program that reads a **Fahrenheit** degree in double from console, then converts it to **Celsius**.

The formula for the conversion is as follows:

$$\text{Celsius} = (5/9) (\text{Fahrenheit} - 32)$$

```
import java.util.Scanner;
public class Fahrenheit_pro {
public static void main(String args[]) {
Scanner input =new Scanner(System.in);
System.out.print("Enter a degree in Fahrenheit: ");
double fahrenheit = input.nextDouble();
double celsius = (5.0 / 9) * (fahrenheit - 32);
System.out.println("Fahrenheit " + fahrenheit + " is " +
celsius + " in Celsius"); }}
```

//out

Enter a degree in Fahrenheit: 100

Fahrenheit 100.0 is 37.7777777777778 in Celsius

تمارين الفصل الأول

التمرين الأول:

أكتب برنامجاً يقوم بطباعة النص التالي : (welcome to java)

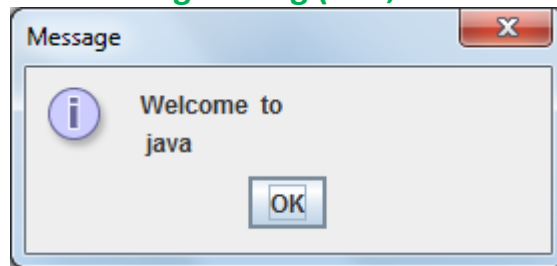
الحل:

```
public class Firstjava
{
    public static void main(String args[])
    {
        System.out.println("welcome to java");
    }
}
```

// Output welcome to java

الحل: (GUI) (Graphical User Interface) واجهة المستخدم الرسومية

```
import javax.swing.*;                      // JOptionPane * بدل
public class FirstJava
{ public static void main(String args[])
    {      JOptionPane.showMessageDialog (null,"Welcome to \n"+"java" ); }    }
```



ملاحظة: يجب أن يتطابق اسم ملف java مع اسم الـ class عند حفظ الملف. (يبدأ بحرف كبير)

كل برنامج يحتوي على الأقل على method main()

ملاحظة: Swing in Java عبارة عن مجموعة أدوات واجهة المستخدم الرسومية تحتوي على مجموعة متنوعة من الأدوات لتصميم تطبيقات محسنة تعتمد على النوافذ.

التمرين الثاني:

أكتب برنامجاً يقوم بإظهار نافذة Frame عنوانها (my first window)

الحل: (GUI)

```
import javax.swing.*;
public class Myprogram2 {
    public static void main(String args[]) {
        JFrame mywindow;
        mywindow=new JFrame();
        mywindow.setSize(400,100);
        mywindow.setTitle("my first window");
        mywindow.setVisible(true);
    }
}
```

السطر الأول أساسي في كل code حيث أن import كلمة استدعاء
javax ملف يحتوي كثير من الأوامر ومنها الأمر swing (تأرجح- تقلب)

التمرين الثالث:

اكتب برنامجا يقوم بحساب مساحة دائرة نصف قطرها 2 حيث نصح عن ثابت $PI=3.14$

الحل:

```
public class AreaCircle {
public static void main(String args[]) {
    final double PI = 3.14159;           // Declare a constant
    double radius = 2;
    double area = radius * radius * PI ;
    System.out.println("area= "+ area);
}}
```

Output

area= 12.56636

نفس البرنامج السابق حيث يتم إدخال نصف القطر من قبل المستخدم (console)

الحل:

```
import java.util.Scanner;
public class InputConsole {
    public static void main(String args[]) {
        Scanner input =new Scanner(System.in);    // Create a Scanner object
        System.out.print("Enter a number for radius: ");
        double radius = input.nextDouble() ;
        double area = radius * radius * 3.14159;
        System.out.println("area="+area);
    }
}
```

run:

Enter a number for radius: 3.5

area=38.4844775

نفس المثال السابق باستخدام (GUI)

الحل:

```
import javax.swing.*;
public class Test1 {
    public static void main(String args[]) {

        final double pi=3.14;
        double radius, area;
        String radiusStr;
        radiusStr = JOptionPane.showInputDialog(null,"Enter radius:");
        radius = Double.parseDouble( radiusStr);
        area=pi*radius*radius;

        JOptionPane.showMessageDialog(null,"area = "+area);
    }
}
```

التمرين الرابع:

اكتب برنامجا يقوم بتحويل الثواني إلى دقائق وباقي ثواني أقل من 60.
فمثلا 700 ثانية تحوي 11 دقيقة و40 ثانية (يتم إدخال الثواني من المستخدم)
الحل: (console)

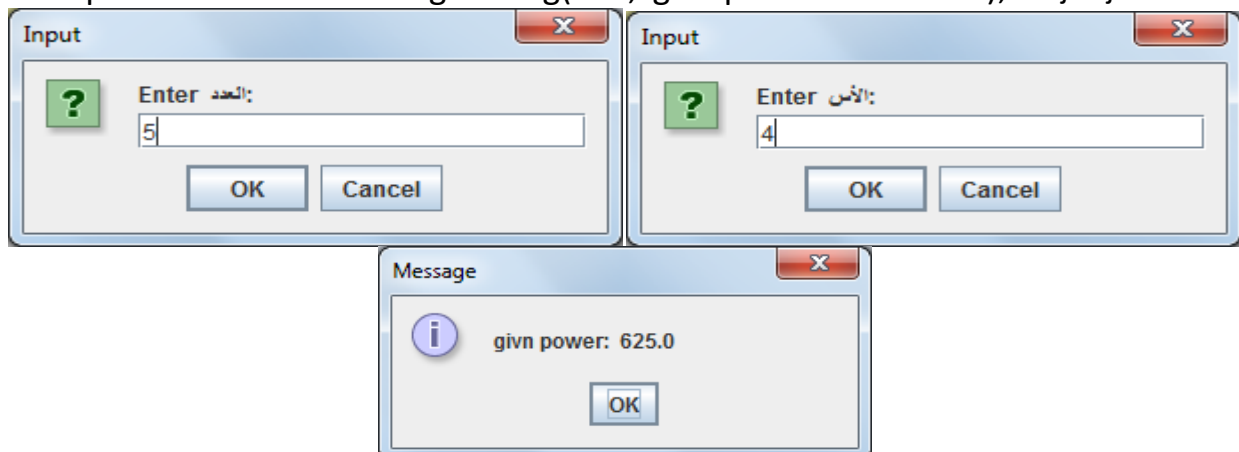
```
import java.util.Scanner;
public class MinutesSeconds {
    public static void main(String args[]) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter an integer for seconds: ");
        int seconds = input.nextInt();
        int minutes = seconds / 60; // حساب الدقائق
        int remainingSeconds = seconds % 60; // الثواني المتبقية
        System.out.println(seconds + " seconds is " + minutes +
            " minutes and " + remainingSeconds + " seconds"); } }
```

Output

Enter an integer for seconds: 700
700 seconds is 11 minutes and 40 seconds

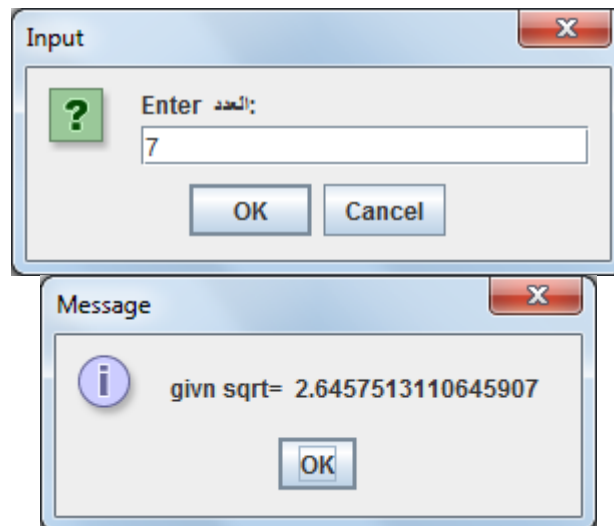
التمرين الخامس: اكتب برنامجا يطلب إدخال العدد والأس لحساب وطباعة ناتج القوة (باستخدام GUI).
الحل:

```
import javax.swing.*;
public class TestMath {
    public static void main(String args[]) {
        String str;
        int num, num1;
        double num2;
        str = JOptionPane.showInputDialog(null, "Enter العدد:");
        num = Integer.parseInt(str);
        str = JOptionPane.showInputDialog(null, "Enter الأس:");
        num1 = Integer.parseInt(str);
        num2 = Math.pow(num, num1);
        JOptionPane.showMessageDialog(null, "givn power: " + num2); } }
```



التمرين السادس: اكتب برنامجا يطلب إدخال عدد ثم يحسب ويظهر الجذر التربيعي له (GUI)
الحل:

```
import javax.swing.*;
public class MathSqrt {
    public static void main(String args[]) {
        String str;
        int num;
        double num1;
        str=JOptionPane.showInputDialog(null,"Enter العدد:");
        num=Integer.parseInt(str);
        num1=Math.sqrt(num);
        JOptionPane.showMessageDialog(null, "givn sqrt= " + num1);
    } }
```



التمرين السابع: اكتب برنامجا يقوم بحساب وتر المثلث القائم حسب فيثاغورس.
الحل:

```
import java.util.Scanner;
public class TrianglePythagoras {
    public static void main(String args[]) {
        Scanner input =new Scanner(System.in);
        System.out.print("Enter a : ");
        double a = input.nextDouble() ;
        System.out.print("Enter b : ");
        double b = input.nextDouble() ;
        double c = Math.sqrt(a*a+b*b);
        System.out.println("c="+c);
    } }
```

run:

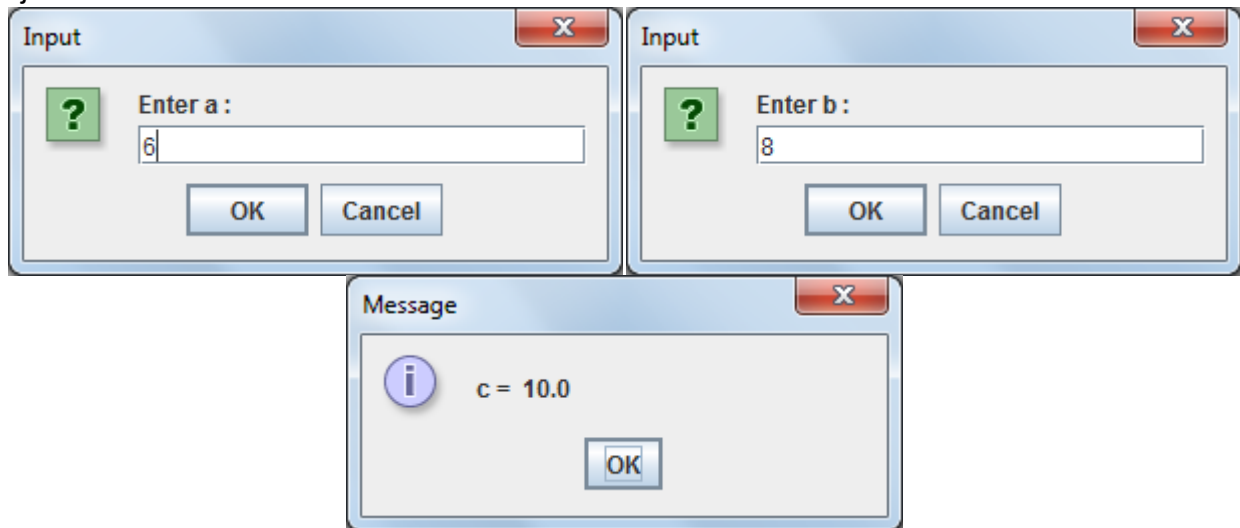
Enter a : 6

Enter b : 8

c=10.0

نفس المثال السابق باستخدام (GUI)

```
import javax.swing.*;
public class TrianglePythagoras {
public static void main(String args[]) {
    double a,b,c;
    String aStr,bStr;
    aStr = JOptionPane.showInputDialog(null,"Enter a :");
    bStr = JOptionPane.showInputDialog(null,"Enter b :");
    a = Double.parseDouble( aStr);
    b = Double.parseDouble( bStr);
    c = Math.sqrt(a*a+b*b);
    JOptionPane.showMessageDialog(null,"c = "+c);
} }
```



ما هو خرج هذه البرامج:

الكود الأول:

```
int a=10;
int b=10;
System.out.println(a++ + ++a);           //10+12=22
System.out.println(b++ + b++);           //10+11=21
//out
22
21
```

الكود الثاني:

```
System.out.println(10*10/5+3-1*4/2);      //21
//out
21
```

الكود الثالث:

```
System.out.println(10<<2);           //10*2^2=10*4=40
System.out.println(10<<3);           //10*2^3=10*8=80
System.out.println(20<<2);           //20*2^2=20*4=80
System.out.println(15<<4);           //15*2^4=15*16=240
//out
40
80
80
240
```

الكود الرابع:

```
System.out.println(10>>2);           //10/2^2=10/4=2
System.out.println(20>>2);           //20/2^2=20/4=5
System.out.println(20>>3);           //20/2^3=20/8=2
//out
2
5
2
```

الكود الخامس:

```
int a=10;
int b=5;
int c=20;
System.out.println(a<b&&a<c);         //false && true = false
System.out.println(a<b&a<c);         //false & true = false
//out
false
false
```

الكود السادس:

```
int a=10;
int b=5;
int c=20;
System.out.println(a<b&&a++<c);       //false && true = false
System.out.println(a);               //10 because second condition is not checked
System.out.println(a<b&a++<c);       //false && true = false
System.out.println(a);               //11 because second condition is checked
//out
false
10
false
11
```

الكود السابع:

```
int a=10;
int b=20;
a+=4; //a=a+4 (a=10+4)
b-=4; //b=b-4 (b=20-4)
System.out.println(a);
System.out.println(b);
//out
14
16
```

الكود الثامن:

```
int a=10;
a+=3; //10+3
System.out.println(a);
a-=4; //13-4
System.out.println(a);
a*=2; //9*2
System.out.println(a);
a/=2; //18/2
System.out.println(a);
//out
13
9
18
9
```