هذه المحاضرة الأولى
برمجة

Computer languages

Unfortunately for us, computers can't understand spoken English ~~of~~ or any other natural language. they can understand directly is **machine code**, which consists of 1s and 0s (binary code).
↓ The only language

Unfortunately : لسوء الحظ
machine code : لغة الآلة
binary code : لغة ثنائية
natural language : لغة طبيعية

Machine code is too difficult to write. For this reason, we use symbolic languages to communicate instructions to the computer. For example, **assembly languages** use abbreviations such as ADD, SUB, MPY to represent instructions.

Symbolic languages : لغات الترميز
communicate : إيصال
instructions : تعليمات

Use	
• action that **might** have taken place in the past	if sentences type III (*If I had seen that, I would have helped.*)

tense	Affirmative / Negative / Question
Conditional II Progressive	**A:** He would have been speaking. **N:** He would not have been speaking. **Q:** Would he have been speaking?

Use
• action that might have taken place in the past • puts emphasis on the **course / duration** of the action

انتهت المحاضرة

assembly language: لغات التجميع

abbreviations: اختصارات

The program is then translated into machine code by a piece of software called an **assembler**. Machine code and assembly languages are called **low-level languages** because they are closer to the hardware. They are quite complex and restricted to particular machines. To make the programs easier to write, and to overcome the problem of intercommunication between different types of computer, software developers designed **high-level languages**, which are closer to the English language.

Piece of Soft ware : قطعة من البرمجيات

assembler : المجمع

low-level Languages: لغات منخفضة المستوى

hard ware : الجزء الصلب (عتاد الحاسب)

complex : معقدة

restrited : مقصورة

particular machines : آلات خاصة

overcome : التغلب

intercommunication : التواصل

developers : مطورو البرمجيات

designed : صمموا

Here are some examples:

- **FORTRAN** was developed by IBM in 1954 and is still used for scientific and engineering applications.

الترجمة:

scientific : علمية

engineering : هندسة

applications : تطبيقات

- **COBOL** (Common Business Oriented Language) was developed in 1959 and is mainly used for business applications.

ترجمة

غالباً - الى حد بعيد

used for business applications: مستخدمة على نطاق واسع

(Common Business Oriented Language): لغة المشتركة الموجهة للشركات

- **BASIC** was developed in 1959 and was widely used in microcomputer programming because it was easy to learn. **Visual BASIC** is a modern version of the old BASIC language, used to build graphical elements such as buttons and windows in Windows programs.

microcomputer : الحواسيب الصغيرة

modern version : نسخة حديثة

to build : لبناء

graphical elements : عناصر رسومية

buttons : الأزرار

windows : النوافذ

أي :

(تستخدم لبناء تطبيقات رسومية Visual BASIc)

- **PASCAL** was developed in 1971. It is used in universities (جامعات) to teach the fundamentals of programming.

fundamentals of programming : أساسيات البرمجة

- **C** was developed in 1980s at AT&T. It is used to write system software, graphics and commercial application. **C++** is a version of C which incorporations object-oriented programming: the programmer concentrates on particular things (a piece of text, a graphic or table, etc.) and gives each object functions which can be altered without changing the entire program. For example, to add a new graphics format, the programmer needs to rework just the graphics object. This makes programs easier to modify.

System software : برمجيات النظام (أنظمة التشغيل)

graphics : رسومية

commercial application : تطبيقات تجارية

version : إصدار

incorporation : التكامل

object_oriented : programming : البرمجة غرضية التوجه

programmer : المبرمج

particular things : أشياء خاصة

graphic or table : شكل أو جدول

object : كائن

functions : وظائف

altered : تغييرها

entered program : كامل البرنامج

new graphics : رسم جديد

rework : إعادة صياغة

modify : تعديل

تطبيقات

- Java was designed by Sun in 1995 to run on the Web. Java applets provide animation and interactive features on Web pages.

run on : تشغيل

animation : حركية - متحركة

interactive : تفاعلية

features : أشكال

Java applets : (صغيرة لبرمجيات الجافا أو برامج للجافا)

تطبيقات

- Programs written in high-level languages must be translated into machine code by a **compiler** or an **interpreter**. A compiler translates the source code into **object code** – that is, it converts the entire program into machine code in one go. On the other hand, an interpreter translates the source code line by line as the program is running.

translated : تُترجَم

compiler : مترجم

interpreter : مفسر

Source code : النص المصدري

object cod : نص كائن

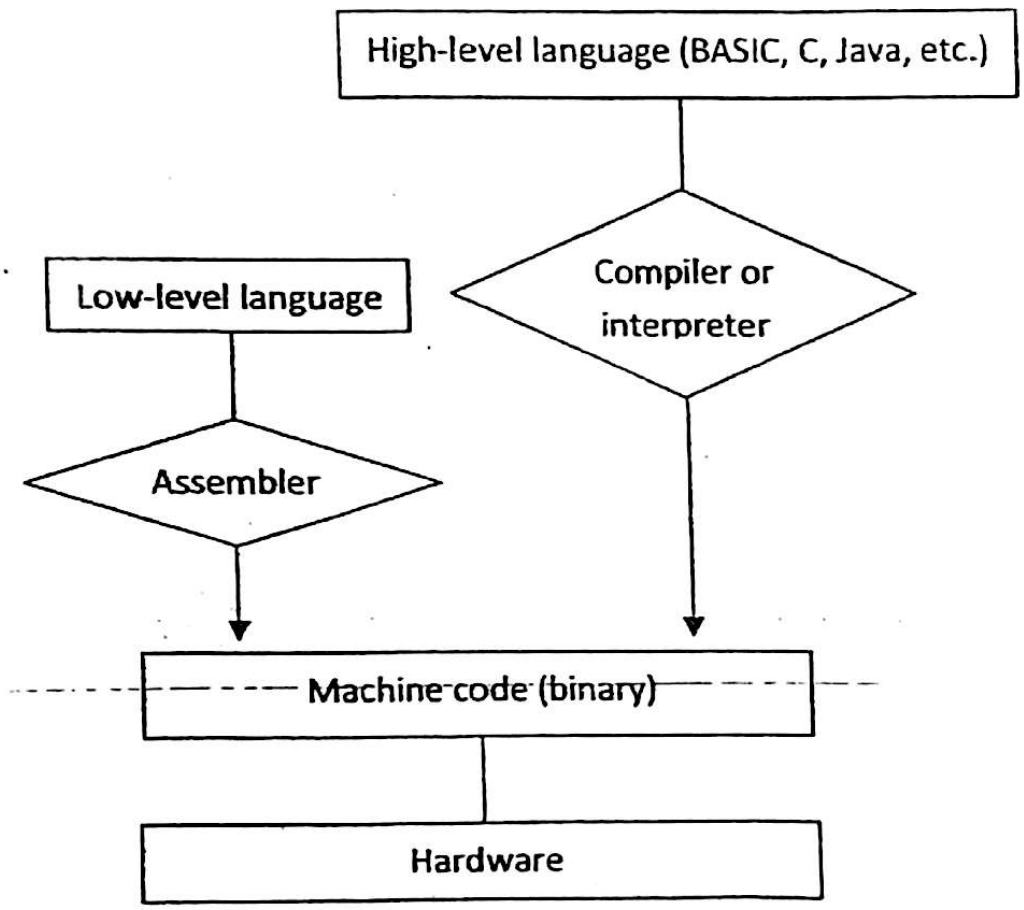
converts : ينقل

entire Program: البرنامج الكامل:

machine code : لغة الآلة

ملاحظة :

(المترجمين يميلون إلى استخدام اللغات ذات المترجم)



It is important not to confuse **programming languages** with **markup languages**, used instructions, known as **markup tags**, to format and link text files. Some examples include:

وسوم
لطاقات

not to confuse : عدم الخلط

Programming Languages: لغات البرمجة

mark up languages : لغات الترميز

used instructions : تعليمات الاستخدام

mark up tags : واسمات الترميز

format : تشكيل

link : ربط

text files : ملفات النصوص

- **HTML**, which allows us to describe how information will be displayed on web pages.

displayed : عرض

- **XML**, which stands for **E**xtensible **M**arkup **L**anguage. While HTML uses pre-defined tags. XML enables us to define our own tags; it is not limited by a fixed set of tags.

Stands : تمثل

Extensible Markup Language : لغة الترميز الموسعة

uses pre-defined tags : واسمة معرفة مسبقاً

enables : تمكن

define : تعريف

fixed set of tags : المجموعة المحددة من الواسمات

- **VoiceXML**, which makes Web content accessible via voice and phone. VoiceXML is used to create voice applications that run on the phone, whereas HTML is used to create visual applications (for example, webpages).

Web content محتوى الويب

used instructions : تعليمات الاستخدام

markup tags : واسمات الترميز

format : تشكيل

link : ربط

text files : ملفات النصوص

- **HTML**, which allows us to describe how information will be displayed on web pages.

displayed : عرض

- **XML**, which stands for **E**xtensible **M**arkup **L**anguage. While HTML uses pre-defined tags. XML enables us to define our own tags; it is not limited by a fixed set of tags.

Stands : تمثل

Extensible Markup Language : لغة الترميز الموسعة

uses pre-defined tags : واسمة معرفة مسبقاً

enables : تمكن

define : تعريف

fixed set of tags : المجموعة المحددة من الواسمات

- **VoiceXML**, which makes Web content accessible via voice and phone. VoiceXML is used to create voice applications that run on the phone, whereas HTML is used to create visual applications (for example, webpages).

Web content محتوى الويب

accessible via voice : متاح بواسطة الصوت

voice application تطبيقات صوتية

run on تُشغّل

visual application تطبيقات مرئية

```
<xml>
<name> Andrea Finch </name>
<homework> Write a paragraph describing the C language </homew>
</xml>
```

in this XML example we have created two new tags:<name> and <homework>

ملاحظة : كل شيء بين < > هذه واسمه (علامة) (تاغ)

The Java language

Java is programming language developed by Sun Microsystems, specially designed to run on the Web. Java programs (call **applets**) let you watch animated characters and moving text, play music. and interact with information on the screen (for example, control animations and select options).

Sun Microsystems : اسم شركة

animated : متحركة

characters : شخصيات

moving text : نصوص متحركة

control: التحكم

animations رسوم متحركة

select : تحديد

(call applets) جزء ثاني للجافا

options : الخيارات

interact: التفاعل

Characteristics of the Java language

Java is an **object-oriented** language, similar to C++, but more dynamic and simplified to eliminate possible programming errors. A Java program is both compiled and interpreted. First, the source code (a file with a **.java** extension) is compiled and converted into a format called bytecode (a file with a **.class** extension), which can then be executed by a Java interpreter. Compiled Java code can run on most computers because there are Java interpreters, known as **Java Virtual Machines**, for most operating systems.

Characteristics : خصائص

object-oriented Language : لغة غرضية التوجه

dynamic : فعالية

simplified: بساطة

to eliminate: للتخلص

possible programming errors :
الأخطاء البرمجية المحتملة

compiled : يترجم

interpreted: يفسر

Source code : النص المصدري

(a file with a .java extension) . Java الملف ذو الاحقة

converted : يحول

byte code : رمز ثنائي

executed : ينفذ

interpreter مفسر الجافا

Java Virtual Machines : آلة الجافا الافتراضية

Operating System : أنظمة التشغيل

Java is **multi-threaded**, meaning a Java program can have multiple threads (parts)-that is, many different things processing independently and continuously. This enables the program to make the best use of available CPU power.

multi-threaded : متعددة المسارات (الروابط)

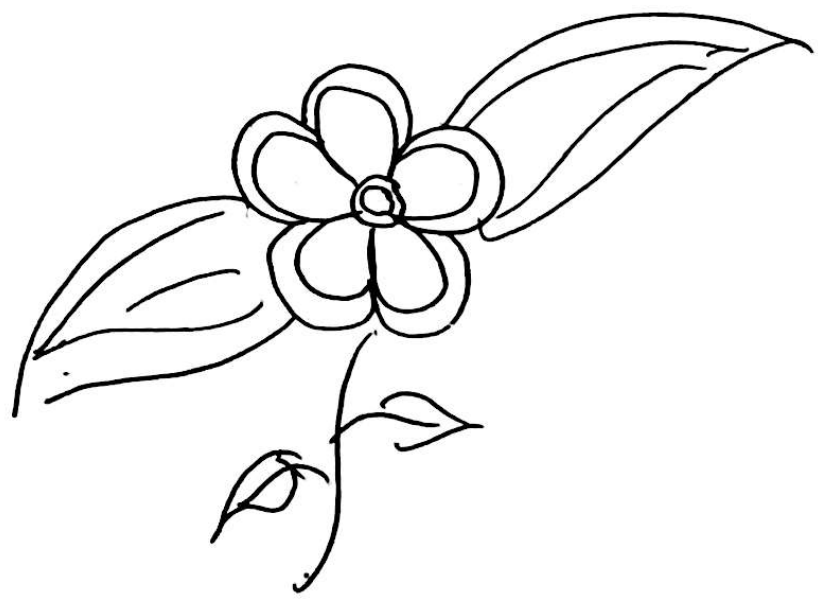
multiple threads : عدة مسارات

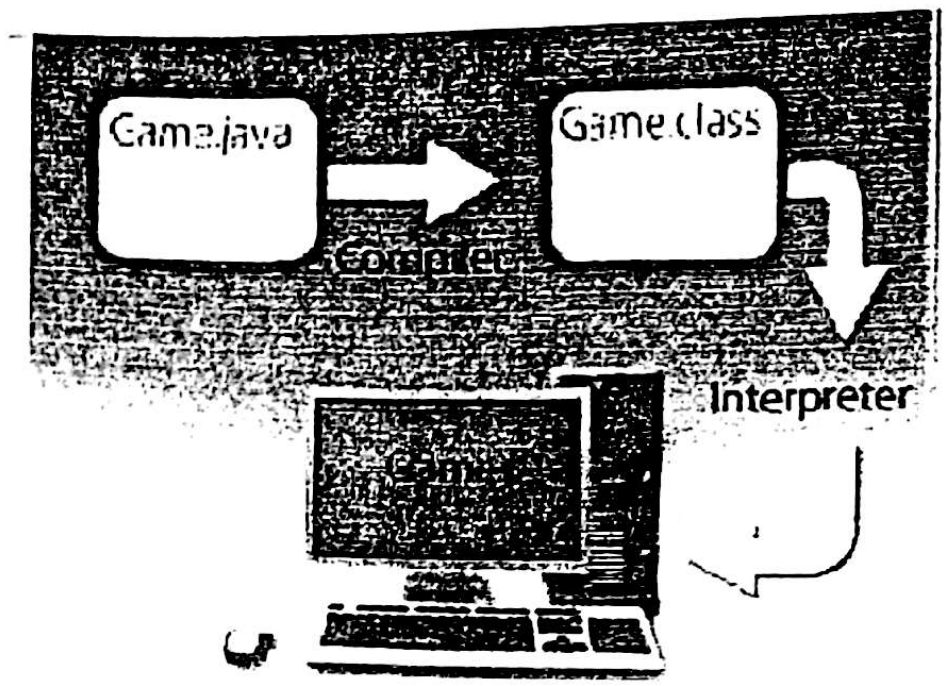
Processing: معالجة

independently : بشكل مستقل

continuously : متتابع

available CPU power: طاقة وحدة المعالجة المركزية المتوفرة (الخامة)





Why is Java popular?

Most programmers like Java because it allows them to write applets which make web pages more interactive and attractive. They can create graphical objects (for example, bar charts and diagrams) and new controls (for example, check boxes and push buttons with special properties). A web page that uses Java can have sounds that play in real time, music that plays in the background, cartoon-style animations, real-time video and interactive games.

interactive : تفاعلية

attractive : جاذبية

graphical objects : كائنات رسومية

bar charts : شريط رسومي

diagrams : رسوم بيانية

controls : أدوات تحكم

check boxes : مربعات اختيار

push buttons : أزرار ضغط

special properties : خصائص خاصة

real time: الزمن الحقيقي

back ground: الخلفية

The Java Micro Edition platform (**Java ME**) is used in mobile devices. It provides flexible tools to create applications that run on mobile phones, PDAs, TV set-top boxes and printers. Nowadays, most phones are configured to use Java games.

Edition platform

(Java ME): بيئة لنوع من الجافا

mobile devices: الأجهزة المحمولة

provides: توفّر

flexible tools: أدوات مرنة

mobilephones: الهواتف النقالة

Tv set-topboxes: أجهزة التلفاز الرئيسية لاستعراض صفحات الويب

printers: الطابعات

configured: معدة

PDAs: (Personaldigital asystem)

Alternatives to Java

One alternative to Java is Microsoft's **C#**, pronounced 'C sharp', a **.NET** language based on C++ with elements from Visual BASIC and Java. There are no substantial differences between C#/Java. When software developers do measurements on pieces of code, sometimes Java is faster, sometimes C# is.

Alternatives to Java بدائل عن الجافا

based on : أساسا

elements : عناصر

substantial : جوهرية

defferences : فروق

developers : مطوري البرنامج

measurements on pieces of cod : قياسات على أداء البرامج

Another competitor is Adobe **Flash** technology. which supports graphics, a scripting language called ActionScript, and the streaming of audio and video. Flash is used to create animation and advertisements, to integrate video into web pages, and to develop rich internet applications such as portals. **Flash files**, traditionally called flash movies, have a **.swf** file extension. They may be an object on a web page or played in the stand alone flash player.

competitor : منافس

technology : تقنية

supports graphics : تدعم الرسوميات

ascriptingLanguage : لغة الكتابة

streaming : تدفق

audio : الصوت

animation : حركات

advertisements : اعلانات

Flash files : ملفات فلاش

video : أفلام

develop : تطور

played : تشغل

portals : البوابات

stand alone : بشكل مستقل

flash player : مشغل فلاش

integrate : تدمج

The End

Introduction to Programming in Java

Ihsan Rehman

This tutorial is intended to introduce the features of the basic programming concept of computer science using Java. This tutorial also provides an overview of the Java language and the facilities that are provided by the standard Java class libraries, including the Java *Application Programmer Interface* (API) class libraries.

tutorial: برنامج تعليمي

intened : يهدف

introduce : ادخال

computer science using Java
علم الحاسب باستخدام الجافا

overview نظرة عامة

the features of the basic program concept : المفاهيم الأساسية لمفهوم البرمجة

facilities : التسهيلات

Java class libraries
مكتبات صفوف الجافا

Application Program Interface (API) class libraries
استدعاء واجهة التطبيق البرمجية .

The tutorial is not intended to completely cover the Java language and facilities, but rather to provide enough understanding of the basic concepts to allow a student to more easily look up additional and more specific information as needed.

لا يقصد بهذا البرنامج التعليمي تغطية كاملة للغة Java وتسهيلاتها بل عوضاً عن ذلك هو يزود بفهم كاف للمفاهيم الأساسية التي تسمح للدارس بسهولة البحث عن المزيد إضافة إلى معلومات أكثر تحديداً حسب الحاجة .

Computer science is not just about programming techniques. It rests on deep ideas and the nature of computation. I want students to understand these deep ideas, as well as grasp the practicality of computation and experience the pleasure of computing.

علوم الحاسب هي ليست فقط تقنيات الحاسب ، هي تقوم على مفاهيم عميقة (أكثر عمقاً) وطبيعة الحاسب ، المؤلف يريد من الدارسين فهم هذه المفاهيم العميقة فضلاً عن ذلك فهم عملية الحساب وتجربة متعة الحسابات .

Types of Programs:

There are two categories of programs. *Application programs* (usually called just "applications") are programs that people use to get their work done. Computers exist because people want to run these programs. *Systems programs* keep all the hardware and software running together smoothly.

Application Programs:
- Word processors
- Game programs
- Spreadsheets
- Graphics programs
- Web browsers

Systems Programs
- Operating system.
- Networking system.
- Web site server.
- Database system.
- Programming language software.

JAVA

The most important systems program is the *operating system*. The operating system is always present when the computer is running. It coordinates the operation of all the hardware and software components of the computer system. The operating system is responsible for starting application programs running and finding the resources that they need..

Application programs: برامج تطبيقية (تطبيقات)

System Programs: أنظمة التشغيل (البرامج)

resources: الموارد (الموارد)

Executable program or Machine Language program:

Each different type of CPU has its own unique language that it understands, called machine language. Every program must be converted into a machine language before it executes. A program in the form of machine language is also called executable program.

Executable program: البرامج القابلة للتنفيذ

Machine Language program: برامج لغة الآلة

converted: تحول

The tutorial is divided into seven chapters. Each chapter contains three parts of the lesson

1. Lesson
1. Review questions
2. Programming exercises (Labs)

The tutorial is also an excellent supplement for the ICS students.

A 75 mints log sample test which covers all the materials from chapter 1 to chapter 7 is included at the end of the tutorial.

You are welcome to e-mail me at ihsan1@gmail.com with your thoughts and comments while you are reading this tutorial or afterwards. I promise to get back to you speedily.

Chapter 1 – Basic Computer Science Concepts & Java

Computer:

A computer is a complex machine that executes software (programs, data). It consists of many different components. But at the heart -- or the brain, if you want -- of the computer is a single component that does the actual computing. This is the *Central Processing Unit*, or CPU. In a modern desktop computer, the CPU is a single "chip" on the order of one square inch in size. The job of the CPU is to execute programs.

Programs:

Programs are lists of instructions for the CPU or Processor to perform a task.

Software:

Software consists of both *programs* and *data*. Data can be any information that a program needs: character data, numerical data, image data, or many other types.

executes : ينفذ

software : برمجيات

chip : شريحة

Processor : معالج

perform a task : أداء مهمة

Example of machine language program and its execution:

Let us say that an electric toothbrush has a CPU (Processor). The processor can rotate the bristles left and right, and can check the on/off switch. Here is imaginary table for machine instructions (Machine Language) and machine operation.

Machine Instruction	Machine Operation
0000 0000	Stop
0000 0001	Rotate bristles left
0000 0010	Rotate bristles right
0000 0100	Go back to start of program
0000 1000	Skip next instruction if switch is off

Programming Languages:

Programmers write programs using some languages that called programming languages. There are two types of programming languages.

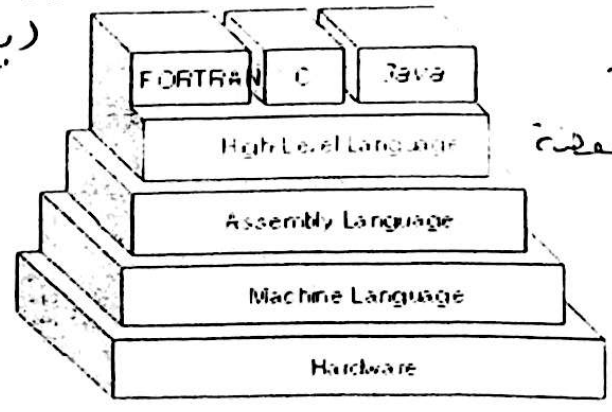
1. Low Level Languages
2. High Level Languages

Low Level Languages:

A low-level programming language is a language that provides little or no abstraction from low-level CPU operations.

High Level Languages:

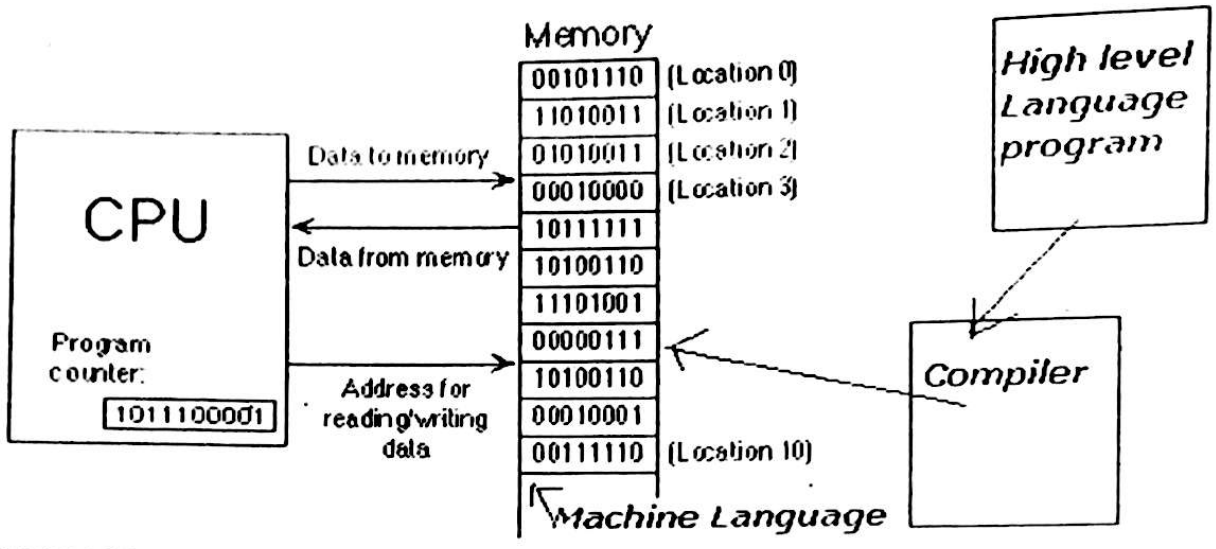
A high-level programming language is a programming language that is easier to program in, to some extent platform-independent like Java, and abstract from low-level CPU operations.



Compiler:

As we read in previous pages, a computer perhaps CPU or processor can understand only machine language. And programmers prefer to write their programs in high level languages because it is easy to write and human readable.

A compiler is a program that translates a high level language to a low-level machine



language.

Java (A Programming Language):

Java is a high-level, object-oriented programming language developed by Sun Microsystems. It is similar to C++, but has been simplified to eliminate language features that cause common programming errors. Java is a general purpose programming language with a number of features that make the language well suited for use on the Web and it has a very rich library, called Java API. We will explore these features in later lessons.

Of course to write and run a Java program you need *Java Development Kit* (JDK). The *Java Development Kit* (JDK) is a collection of software for developing Java programs available at no charge from Sun Microsystems.

object-oriented : غرضية التوجه

Programming Language: لغة البرمجة

Installing JDK:

JDK download is available at java.sun.com. Detailed instructions on how to download it and install it are available at that site. If you have not yet installed Java, get the most recent version. The JDK for Windows comes in a file named `jdk1.5.0_01-win.exe` (or a similar name.)

Edit – Compile – Run, Java Programs:

Here is an example Java program. Some of its details will be explained later. This program will be created as a text file using any editor such as the "NotePad" editor. The

file that contains this program should be named `Hello.java`. This file is called a *source program*.

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

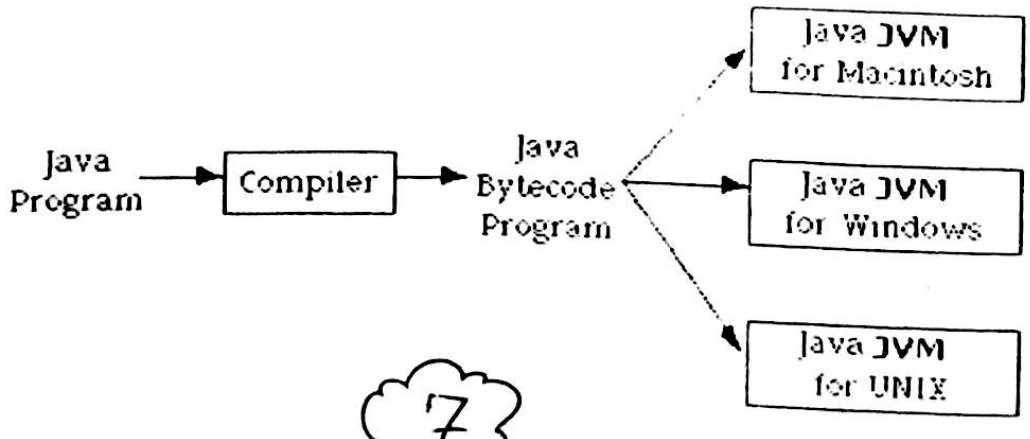
Java compiler converts this source program (Hello.java) into Java bytecodes (Hello.class). Then an other program called Java Virtual Machine JVM converts these bytecodes into machine language.

Java Virtual Machine:

You might be thinking why we use JVM. Why we do not convert source code like Hello.java directly into machine language. There are many reasons for that.

1. **Platform Independent:**

   Java bytecodes file can be run on any platform. The only we need JVM and java bytecodes.



Source program: ملف / برنامج مصدري

platform Independent : استقلالية المنصة

platform: منصة

2. Security:

Many Java programs are to be downloaded over a network. This leads to obvious security concerns: you don't want to download and run a program that will damage your computer or your files. The JVM acts as a buffer between you and the program you download. You are really running the JVM, which runs the downloaded program indirectly. The JVM can protect you from potentially dangerous actions on the part of that program.

3. Java Applets:

An applet is a Java bytecode program that runs on a Web browser. Often applets are used for complicated user interaction, or for graphics and animation that can't otherwise be done in a Web page. Applets will be discussed in later in these notes.

the downloaded program indirectly : صح البرنامج المحمل بشكل غير مباشر

applet : ملف برنامج الترميز الثنائي

Chapter 2 – Introduction to Java Programming

Here is the source program (source file) similar from the previous day but little advance. The purpose of this program is to do some calculation and putting the result on the monitor.

Example 2.1:

```
1   Public class Area
2   {
3     public static void main(String[] args)
4     {
5        int width;
6        width = 8;
7        int height = 3;
8        int area = width * height;
9        System.out.println(area);
10    }
11  }
```

We start our exploration by looking at the program. This program exposes features common to all Java programs.

Class Definition:

The statement on line 1 indicates the beginning of a class named Area. This statement is called the class header and is followed by two braces (an opening brace on line 2 and closing brace on line 11) that enclose the class body.

Method Definition:

The statement on line 3 indicates the beginning of the main method. This statement is a method header and followed by two braces that encloses the body of the method.

Statements:

The body of the main method contains five statements. A variable name width is declared as an integer (int) on line 5 and is assigned the value 8 on line 6.
Similarly on line a variable hight is declared and assigned the value 3. We can aslo write this line as

```
int height;
height = 3;
```

However, it is convenient and commonly used shortcut to combine declaration with assignment. The statement on line 8 uses same shortcut to declare a third integer variable,

area and assign it the product of the previous two variables. The last statement on line 9 put the area on the screen.

Comments:

Comments are non-executable notes that we add in program to remember what this program or statement do. A comment starts with the two characters "//" (slash). The Java compiler ignores those characters and everything that follows them on that one line. If we need comments more than two lines then we use /* and */. The Java compiler ignores every thing in between these two characters.

Here is Example 2.1 with comments.

Example 2.2:

```
1
2   /* This class calculate the area of a rectangle.
3      Author:  Ihsan Rehman
4      Version: 2.01
5   */
6   Public class Area
7    { // main method
8      public static void main(String[] args)
9      {
10        int width;              // width of rectangle
11        width = 8;
12        int height = 3;         // height of rectangle
13        int area = width * height;
14        System.out.println(area);
15     }
16  } // end of class
```

Language Elements:

Correctness of a sentence written in human language like English is a multistage process. In the first stage, we examine each word in the sentence and attempt to classify it according to the rules of the language, for example, as a verb, noun, proper name, preposition and so on. Similarly, the first stage in determining the correctness of a Java program is to break each of its statements into tokens and to classify each token as one of the lexical element of Java.

In this section, we will look at lexical elements of Java: keywords, identifiers, operators, and separators.

ترجمة:

صحة هذه العبارة المكتوبة باللغة البشرية كاللغة الإنكليزية هي عملية متعددة المراحل

attempt to : يحاول

classify : تصنيف

stage : المرحلة

according to the rules of the language
وفقاً لقواعد اللغة

break : تقطيع

determining the correctness of Java Program
تحديد صحة برنامج الـ Java

tokens : مفردات

Lexical : معجمي (أو علاقة بمفردات اللغة)

1. Keywords:

Keywords are words that have predefined meaning in Java. Some commonly used keywords are public, class, boolean, return, if, break etc.

2. Identifiers:

Identifiers are names that programmers choose to use in their programs; they have no predefined meanings in Java. For example, Area (a class name), main (a method), and width (a variable name) are identifiers.

3. Literals:

Literals are constant values. For example, the number 3 that appears on line in Example 2.1 is an integer literals. Also true, false, null are literals.

4. Operators:

An operator is like a mathematical function because it takes one or more arguments and produce a result. The multiplication asterisk (*) on line 8 in Example 2.1 is an operator that takes two arguments, width and height, and produce a result. Some example of Java operators are = , !, &, % etc

5. Separators:

Separators are characters that separate various part of the Java code. They are similar to punctuation symbols in human language, Java separators are , ; () [] {}

Note: The reserved words are the key words plus the literals: true, false, null

Key words: الكلمة المفتاحية

perdefin : محدد

Identifiers: المعرفات

Literals: ثابت

An operaters: العامل

arguments : وسيط

Separters : الفاصل

characters : محارف

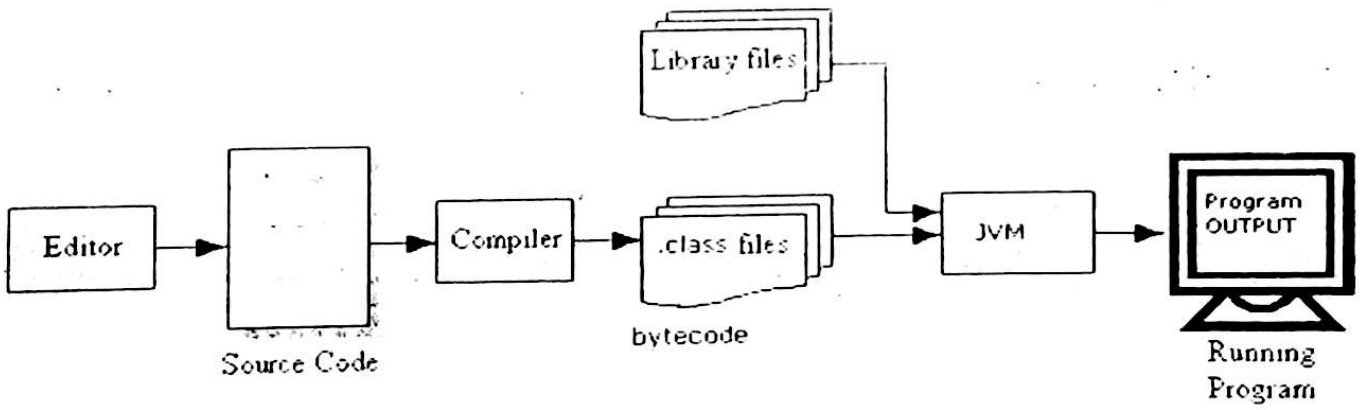
punctuation Symbols
علامات الترقيم

تنفيذ مثال

Program Execution:

Three steps are needed to execute (run) a program:

Edit	We use an editor to write the program and save it in a file. Previous program example 2.2 must be saved in a file name *Area.java* We call this file as a source file.
Compile	We use a compiler to translate our program form Java, a high-level language, to bytecode. The compiler reads the source file, translate it, and stores the translation in a file called the **class file**. The file is so named because it has the same name as the source but with the class extension; for example the source Area.java complies into Area.class. If there is any syntax error in the source file, compiler will issue a list of compile-time errors. In that case, we must go back to the editor, correct the reported errors, save the file and compile again. This process is called edit-compile cycle.
Run	We use a virtual machine JVM to execute the bytecode in the class file.



stdin: ملف ادخال

Chapter 3 – Variables, Data Types and the Declaration Statement

Variable:

A variable is just a memory location (or several locations treated as a unit) that has been given a name so that it can be easily referred to and used in a program. The programmer only has to worry about the name; it is the compiler's responsibility to keep track of the memory location. The programmer does need to keep in mind that the name refers to a kind of "box" in memory that can hold data, even if the programmer doesn't have to know where in memory that box is located.

Data Type:

Every variable in the program has a type that indicates what sort of data it can hold. One type of variable might hold integers -- whole numbers such as 3, -7, and 0 -- while another holds floating point numbers -- numbers with decimal points such as 3.14, -2.7, or 17.0. (Yes, the computer does make a distinction between the integer 17 and the floating-point number 17.0; they actually look quite different inside the computer.) There could also be types for individual characters ('A', ';', etc.), strings ("Hello", "A string can include many characters", etc.), and less common types such as dates, colors, sounds, or any other type of data that a program might need to store.

Variables: المتغيرات

Data Types أنواع البيانات

Declarationstatemenr: عبارات التصريح

referred: يشير

distinection: تمييز

floating point numbers: أعداد الفاصلة العائمة:

Declaration Statement:

The declaration statement has the following form:

type variableName;

In this statement programmer is indicating that a variable with the specified name will be used later in the program and that its future value will be of the specified type. The statement does not assign a value to the variable; it merely prepares it to hold a value of a particular type.

For example, the following statement declares that the programmer intends to use a variable named width and store integer in it.

```
int width;
```

A declaration must appear before the variable is used in the program; otherwise, the compiler will report an error.

We can also initialize a variable by assigning a value to it. This value must be of the correct data type.

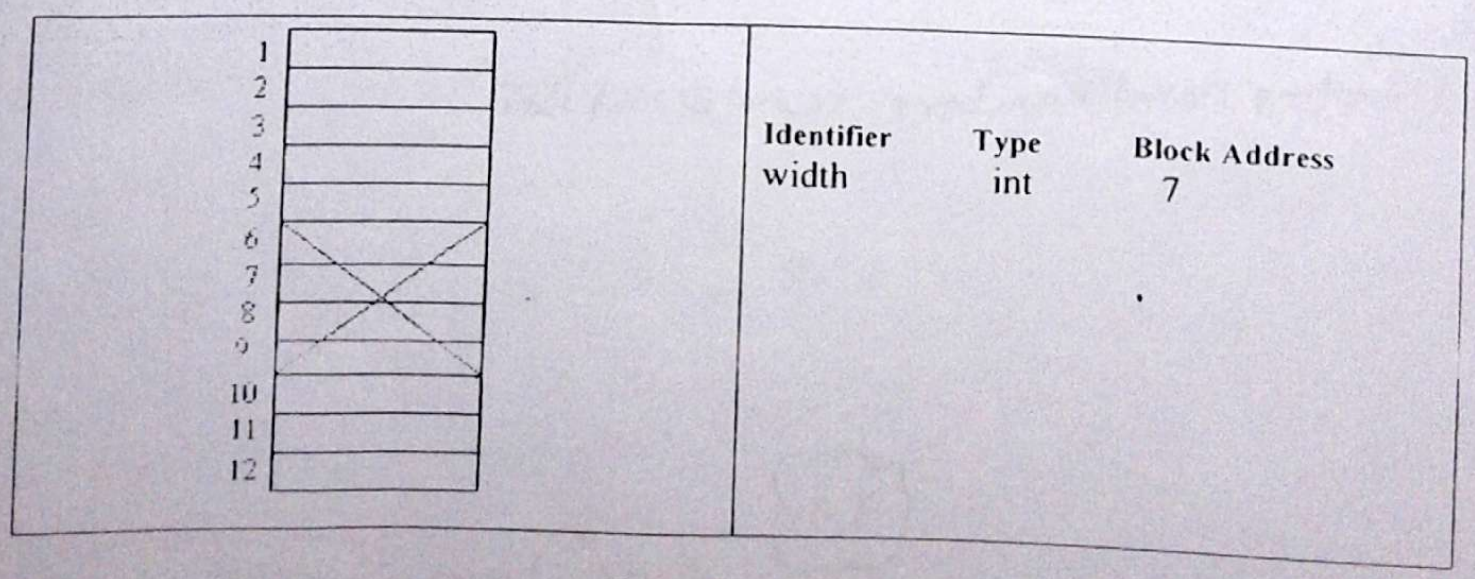
```
int width = 8;
```


Declaration and Memory:

When we declare variable, we are asking the compiler to set aside an area of memory to hold its future values. Memory can be viewed as a one-dimensional arrangement of cells, each of which is called a memory byte. Data type integer takes 4 bytes of the memory. So when the compiler encounters a declaration such that

```
int width;
```

it realize that 4 bytes of memory are needed.



Many Data Types:

Java has very many data types built into it, and you (as a programmer) can create as many more as you want. However, all data in Java falls into one of two categories: primitive data type and non primitive data type (objects).

Primitive Data Types:

Primitive data types are types of data that are fundamental and built into Java. Programmers use these data types to produces other types of data (non-primitive data type). The names of the eight primitive data types are:

byte	short	int	long	float	double	char	boolean

Upper and lower case characters are important. So "byte" is the name of a primitive data type, but "BYTE" is not. Computer languages where case is important are called **case sensitive**. Java is case sensitive language. Some languages are not case sensitive, especially old languages that were designed when data entry equipment did not have lower case characters.

Upper: الأسير

Lower: الصغير

objects: أغراض

Primitive Data typ: أنواع كائنات بدائية

Case sensitive: حالة الحالة

Non Primitive Data Type or Objects:

There are only the eight primitive data types. Any data type you invent will be a type of non-primitive data type or objects. Much more will be said about objects in the future (since Java is a *object oriented* programming language.) The following will be all you need to know, for now:

أعداد محدودة

- A primitive data value uses a small, fixed number of bytes.
- There are only eight primitive data types.
- A programmer can not create new primitive data types.
- An object is a big block of data. An object may use many bytes of memory.
- An object usually consists of many internal pieces.
- The data type of an object is called its **class**.
- Many classes are already defined in the Java Development Kit.
- A programmer can create new classes to meet the particular needs of a program.

Example Program 3.1:

```
public class Example
{
     public static void main(String[] args)
     {
          String name = "Ali";
          double payRate = 10.0;
          double taxRate = 0.10;
          int   hoursWorked = 40;

        System.out.println("Total earning for "+name+":
         "+payRate*taxRate*hoursWorked);

     }
}
```

This example program containing several variable declarations. Variable *name* is of data type String. String is a class coming from Java built in library. Variable payRate is of type double.

The character * means *multiply*. In the program, (hoursWorked * payRate) means to multiply the number stored in hoursWorked by the number stored in payRate. The character + is used for concatenation (connecting two strings).

Output of This Program:

```
Total earning for Ali: 40
```

Example Program 3.2:

```
/*    class Example shows the total earning for Ali.
      Author: Ihsan Rehman
      Copy rights @ 2006
      Version 1.01
*/

public class Example
{
     // main method, holding all data.
           public static void main(String[] args)
     {
          String name = "Ali";     // name
          double payRate , taxRate, totalEarning;
                         int hourWorked;

                         // initialize the variables.
                         payRate =  10.0;
                         taxRate = 0.10;
               hoursWorked = 40;

                         totalEarning =
payRate*taxRate*hoursWorked;    // calculating total

                              System.out.println("Total earning for
"+name+": "+totalEarning);

     }
} // end of class
```

Logically the example program 3.2 is same as Example program 3.1. Only difference is in style. Good programming style is very important, particularly when your program is hundreds lines long.

In the above program, programmer declares the variables and then initializes the variables by assigning them some values. Remember variables can be initialize only with compatible values. It means integer variable can hold only integer values, not characters or any other data type.

Java program tells the computer to take these values stored in these variables, do calculation as program says and then store the result in the variable named "totalEarning" and print the result on the monitor.

sounds fx / ef eks/ may be interpreted as any of: fx, f(x), f_x, FX, FX. The difference is usually made clear by the context; it is only when confusion may occur, or where he wishes to emphasise the point, that the mathematician will use the longer forms: f multiplied by x, the function of x, f subscript x, line FX, vector FX.

Similarly, a mathematician is unlikely to make any distinction in speech (except sometimes a difference in intonation or length of pauses) between pairs such as the following:

$x+(y+z)$	and	$(x+y)+z$
$\sqrt{ax+b}$	and	$\sqrt{(ax+b)}$
a^n-1	and	a^{n-1}

The most common pronunciations are given in the list below. In general, the *shortest* versions are preferred (unless greater precision is necessary).

$x+1$	x plus one
$x-1$	x minus one
$x\pm1$	x plus or minus one
xy	xy / x multiplied by y
$(x-y)(x+y)$	x minus y, x plus y
$\frac{x}{y}$	x over y
$x=5$	x equals 5 / x is equal to 5
$x\equiv y$	x is equivalent to y / x is identical with y
$x>y$	x is greater than y
$x\geq y$	x is greater than or equal to y
$x<y$	x is less than y
$0<x<1$	zero is less than x is less than 1
$0\leq x\leq 1$	zero is less than or equal to x is less than or equal to 1
x^2	x squared
x^3	x cubed
x^4	x to the fourth / x to the power four
x^n	x to the n /x to the nth/ x to the power n
x^{-n}	x to the minus n / x to the power minus n
$\sqrt{x}$	root x / square root x / the square root of x
$\sqrt[3]{x}$	cube root x
$\sqrt[4]{x}$	fourth root x
$\sqrt[n]{x}$	nth root x /'enθ ru:t 'eks/
$(x+y)^2$	x plus y all squared
$\left(\frac{x}{y}\right)^2$	x over y all squared
$n!$	n factorial / factorial n
$x\%$	x per cent /'eks pə 'sent/
∞	infinity
$x\propto y$	x varies as y / x is (directly) proportional to y
$\dot{a}$	a dot /'eɪ dɒt/
$\ddot{a}$	a double dot /'eɪ 'dʌbl dɒt/
$f(x)$	fx /f of x/ the function of x
$f'(x)$	f dash x / the (first) derivative of f with respect to x
$f''(x)$	f double-dash x / the second derivative of f with respect to x

$f'''(x)$	f triple-dash x / f treble-dash x / the third derivative of f with respect to x
$f^{(4)}(x)$	f four x / the fourth derivative of f with respect to x
$\frac{\partial v}{\partial \theta}$	the partial derivative of v with respect to θ
$\frac{\partial^2 v}{\partial \theta^2}$	d two v by d theta squared / the second partial derivative of v with respect to θ
w.r.t.	with respect to
$\int_0^\infty$	the integral from zero to infinity
$\sum_{i=1}^{n}$	the sum from i equals one to n
$\lim_{\Delta x \to 0}$	the limit as delta x approaches zero
$\underset{\Delta x \to 0}{\text{Lt}}$	the limit as delta x tends to zero
grad	gradient
div	divergence
$\log_e y$	log y to the base e / log to the base e of y / natural log (of) y
$l_n y$	log y to the base e / log to the base e of y / natural log (of) y
$\overline{OA}$ or $\overrightarrow{OA}$	OA / vector OA
$x \in A$	x belongs to A / x is a member of A / x is an element of A
$x \notin A$	x does not belong to A / x is not a member of A / x is not an element of A
$A \subset B$	A is contained in B / A is a proper subset of B
$A \subseteq B$	A is contained in B / A is a subset of B
$B \cap A$	B intersection A
$B \cup A$	B union A
cos x	/kɒs eks/
sin x	/saɪn eks/
tan x	/tæn eks/
sec x	/sek eks/
cosec x	/ˈkəʊsek eks/
sinh x	/ʃaɪn eks/ or /sɪntʃ eks/
cosh x	/kɒʃ eks/
tanh x	/θæn eks/ or /tæntʃ eks/
m_a	ma / m subscript a / m suffix a
$x_1 + x_2 + x_3 \ldots\ldots$	(usually) x one plus x two plus x three, etc.
$\lvert x \rvert$	mod x / modulus x

Appendix III

Units

S.I. Units

S.I. Units (S.I. = système international) have now been adopted by most countries of the world and should always be used. Points to note are: