

1.1

لماذا البرنامج؟ : Why Program?

Why Program?

الحاسب: هو آلة قابلة للبرمجة مصممة لتنفيذ وإتباع تعليمات محددة.

البرنامج: هو تعليمات مخزنة في ذاكرة الحاسب تجعله ينفذ شيئاً ما.

المبرمج: هو شخص يكتب تعليمات (برامج) تجعل الحاسب ينفذ مهمة ما.

ولذلك فمن دون المبرمجين ليس هناك برامج وبدون البرامج لا يمكن للحاسب فعل أي شيء.

1.2

أنظمة الحاسب: العتاد الصلب (المكونات المادية)
والبرمجيات

**Computer Systems: Hardware and
Software**

فئات المكونات المادية الرئيسية

Main Hardware Component Categories:

١. وحدة المعالجة المركزية CPU Central Processing Unit
٢. الذاكرة الرئيسية Main Memory
٣. الذاكرة / وحدة التخزين الثانوية / Secondary Memory / Storage
٤. أجهزة الإدخال Input Devices
٥. أجهزة الإخراج Output Devices

Main Hardware Component Categories

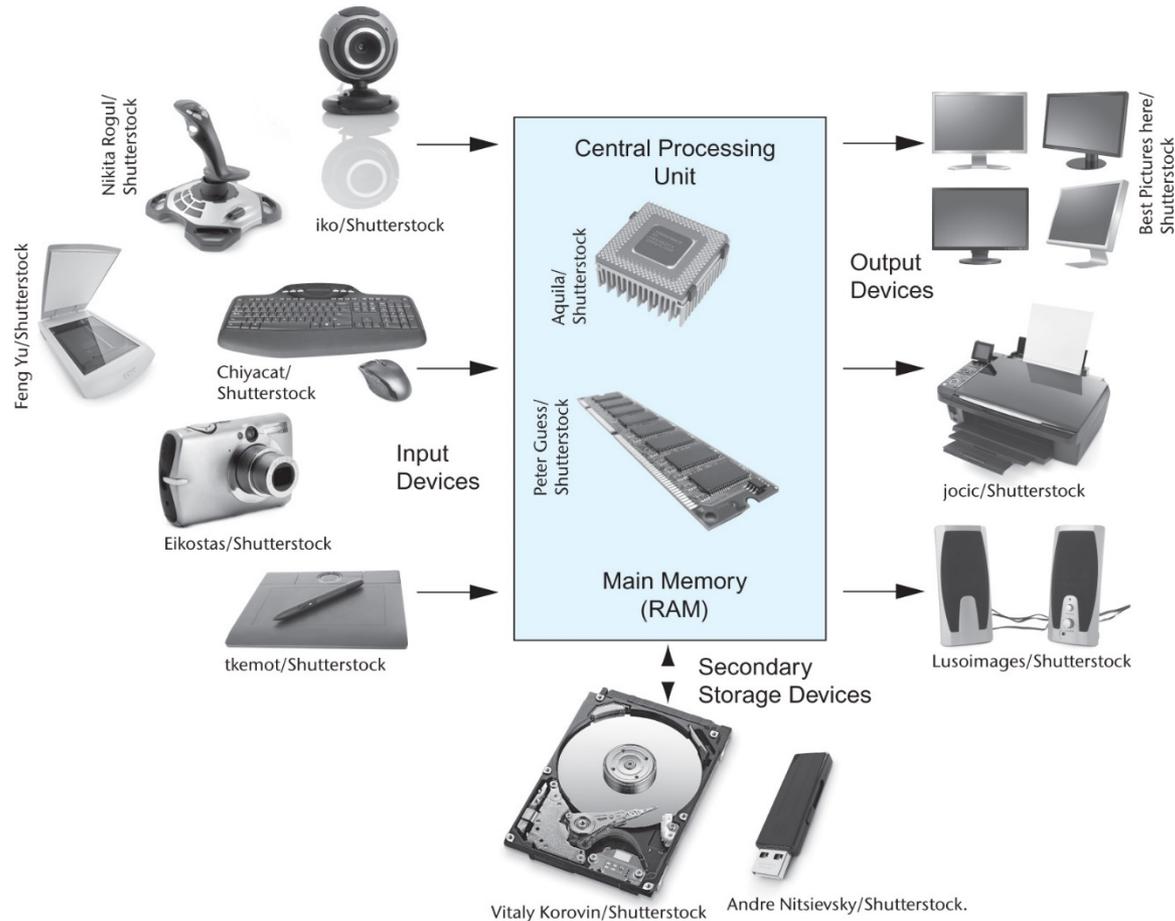


Figure 1-2

وحدة المعالجة المركزية

Central Processing Unit (CPU)

وتضم:

١- وحدة التحكم Control Unit

- تستعيد (من الذاكرة) وتفك شيفرة تعليمات البرامج
- تنسق وتنظم نشاطات (عمليات) كل أجزاء الحاسب الأخرى

٢- وحدة الحساب والمنطق Arithmetic & Logic Unit

- مجموعة من الدارات المصممة بشكل أمثلي لتنفيذ العمليات الرياضية
- ودارات أخرى مصممة لاتخاذ القرارات صح/خطأ ونعم/لا.

تنظيم وحدة المعالجة المركزية: CPU Organization

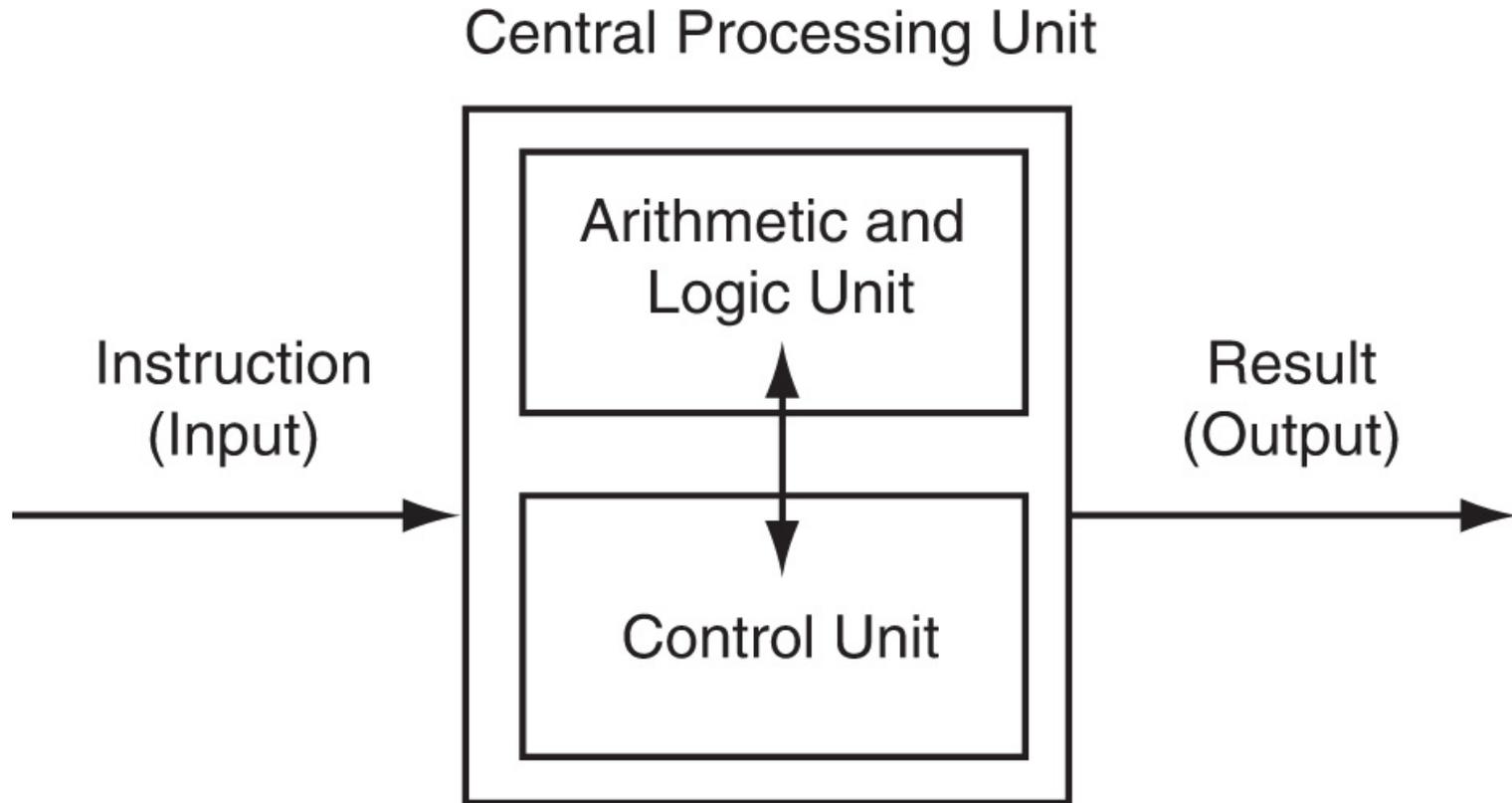


Figure 1-5

الذاكرة الرئيسية: Main Memory

- ذاكرة متطايرة. يتم مسح محتويات الذاكرة الرئيسية عند انتهاء البرنامج أو إطفاء الحاسب.
- تدعى أيضا ذاكرة الوصول العشوائي Random Access Memory (RAM)
- تنظم كالتالي:
 - خانة bit: أصغر قطعة من الذاكرة. تأخذ إما قيمة 0 (تعبّر عن (off, false) أو 1 (on, true)
 - بايت byte: وهي تتالي 8 خانات. للبايتات عناوين.

Main Memory

- العناوين: Addresses يتم تمييز كل بايت في الذاكرة برقم فريد يعرف بالعنوان.

Main Memory

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29

The table shows a grid of memory addresses from 0 to 29. The value 149 is stored at address 16, and the value 72 is stored at address 23.

- In Figure 1-6, the number 149 is stored in the byte with the address 16, and the number 72 is stored at address 23.

وحدات التخزين الثانوية: Secondary Storage units

- غير متطايرة: تحتفظ بالبيانات عند عدم تشغيل البرنامج أو عند إطفاء الحاسب.

- تأتي بأشكال مختلفة الوسائط:

- مغناطيسية magnetic: سواقات الأقراص الصلبة التقليدية التي تستخدم ذراع ميكانيكي قابل للحركة في عمليتي القراءة/الكتابة.

- ذات الحالة الصلبة solid-state: تخزن البيانات في شرائح ولا تحتوي على أجزاء متحركة.

- ضوئية optical: CD-ROM, DVD

- سواقات الفلاش Flash drive: ترتبط مع المنفذ USB

أجهزة الإدخال: Input Devices

- هي وسائل ترسل المعلومات من الخارج إلى الحاسب
- هناك العديد منها:

- لوحة المفاتيح، الفأرة، شاشة اللمس، الماسح الضوئي، الكاميرا
الرقمية، الميكرفون

- سواقات الأقراص HDs، سواقات CD وسواقات DVD

البرمجيات: البرامج التي تعمل على الحاسب

Software-Programs That Run on a Computer

- فئات البرمجيات:

- برمجيات النظام System software: هي برامج تدير مكونات الحاسب المادية وتنفذ عليها.

مثل: أنظمة التشغيل Oses، برامج الخدمات utility prog.s، وأدوات تطوير البرمجيات.

- برمجيات التطبيقات: هي البرمجيات التي تقدم خدمات للمستخدم.

مثل: معالجات الكلمات، الألعاب، البرامج التي تحل مشاكل محددة.

1.3

البرامج ولغات البرمجة

Programs and Programming Languages

Programs and Programming Languages

- البرنامج: هو مجموعة من التعليمات التي يتبعها الحاسب لتنفيذ مهمة ما.
- نبدأ بالخوارزمية *algorithm* والتي هي مجموعة من الخطوات المحددة (المعرفة) جيدا.

مثال: خوارزمية حساب الراتب الإجمالي

Example Algorithm for Calculating Gross Pay

1. Display a message on the screen asking “How many hours did you work?”
2. Wait for the user to enter the number of hours worked. Once the user enters a number, store it in memory.
3. Display a message on the screen asking “How much do you get paid per hour?”
4. Wait for the user to enter an hourly pay rate. Once the user enters a number, store it in memory.
5. Multiply the number of hours by the amount paid per hour, and store the result in memory.
6. Display a message on the screen that tells the amount of money earned. The message must include the result of the calculation performed in Step 5.

لغة الآلة

Machine Language

- مع أن الخوارزمية السابقة تحدد خطوات حساب الراتب الكلي، لكنها غير جاهزة للتنفيذ على الحاسب.
- يستطيع الحاسب تنفيذ تعليمات لغة الآلة machine language فقط

Machine Language

- تعليمات لغة الآلة هي أرقام ثنائية مثل الرقم:

1011010000000101

- بدلا من كتابة البرامج بلغة الآلة يستخدم المبرمجون لغات البرمجة.

البرامج ولغات البرمجة

Programs and Programming Languages

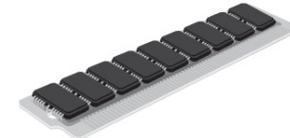
أنواع اللغات:

- منخفضة المستوى: تستخدم للتواصل مع مكونات الحاسب المادية مباشرة. غالبا ما تكتب برامجها بشيفرة الآلة الثنائية (أصفار وواحدات) مباشرة.
- عالية المستوى: أقرب للغة البشر.

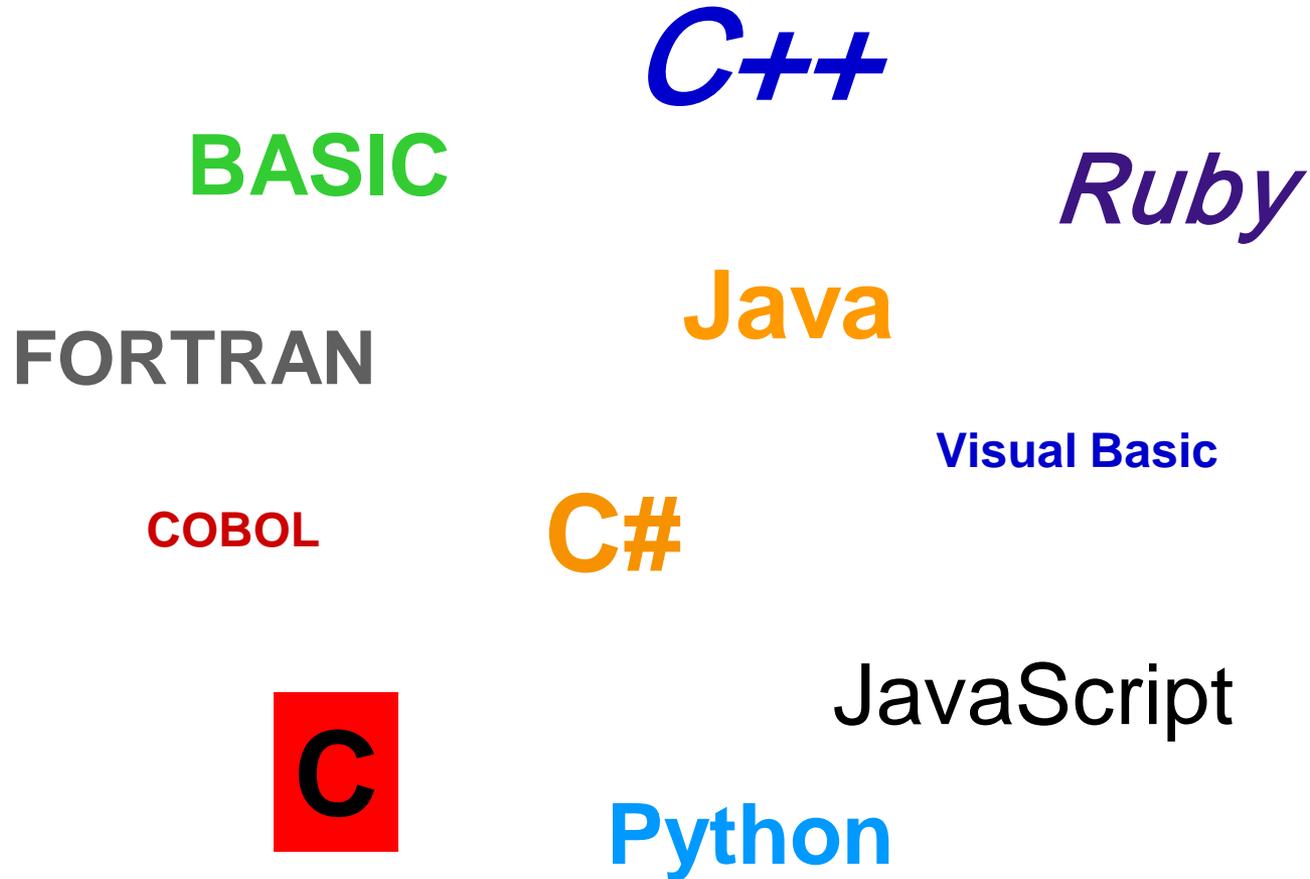
- High level (easily understood by humans)



Low level (machine language)
10100010 11101011



Some Well-Known Programming Languages (Table 1-1 on Page 10)

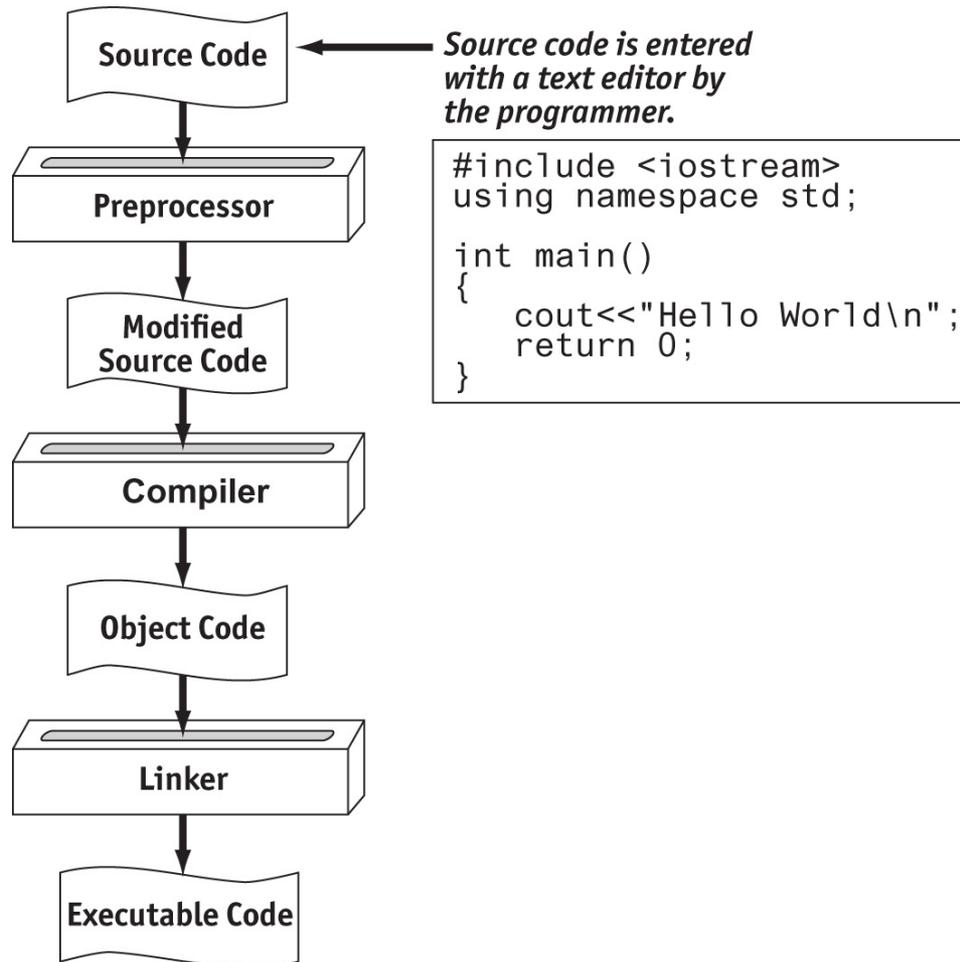


من البرنامج عالي المستوى إلى الملف التنفيذي

From a High-Level Program to an Executable File

- (a) باستخدام محرر النصوص، أنشئ ملف يحتوي على البرنامج.
- (b) شغل المعالج الأولي preprocessor لتحويل موجهات `directives` الملف المصدر إلى عبارات برنامج شيفرة المصدر.
- (c) شغل المترجم compiler لتحويل البرنامج المصدر إلى تعليمات الآلة.
- (d) شغل الرابط linker لوصل الشيفرة الخاصة بالمكونات المادية مع تعليمات الآلة مما يعطي الملف التنفيذي.
- (e) غالبا ما تنفذ الخطوات من b حتى d بأمر واحد أو ضغطة زر.
 - ستمنع الأخطاء المكتشفة في أي خطوة تنفيذ الخطوات التالية.

From a High-Level Program to an Executable File



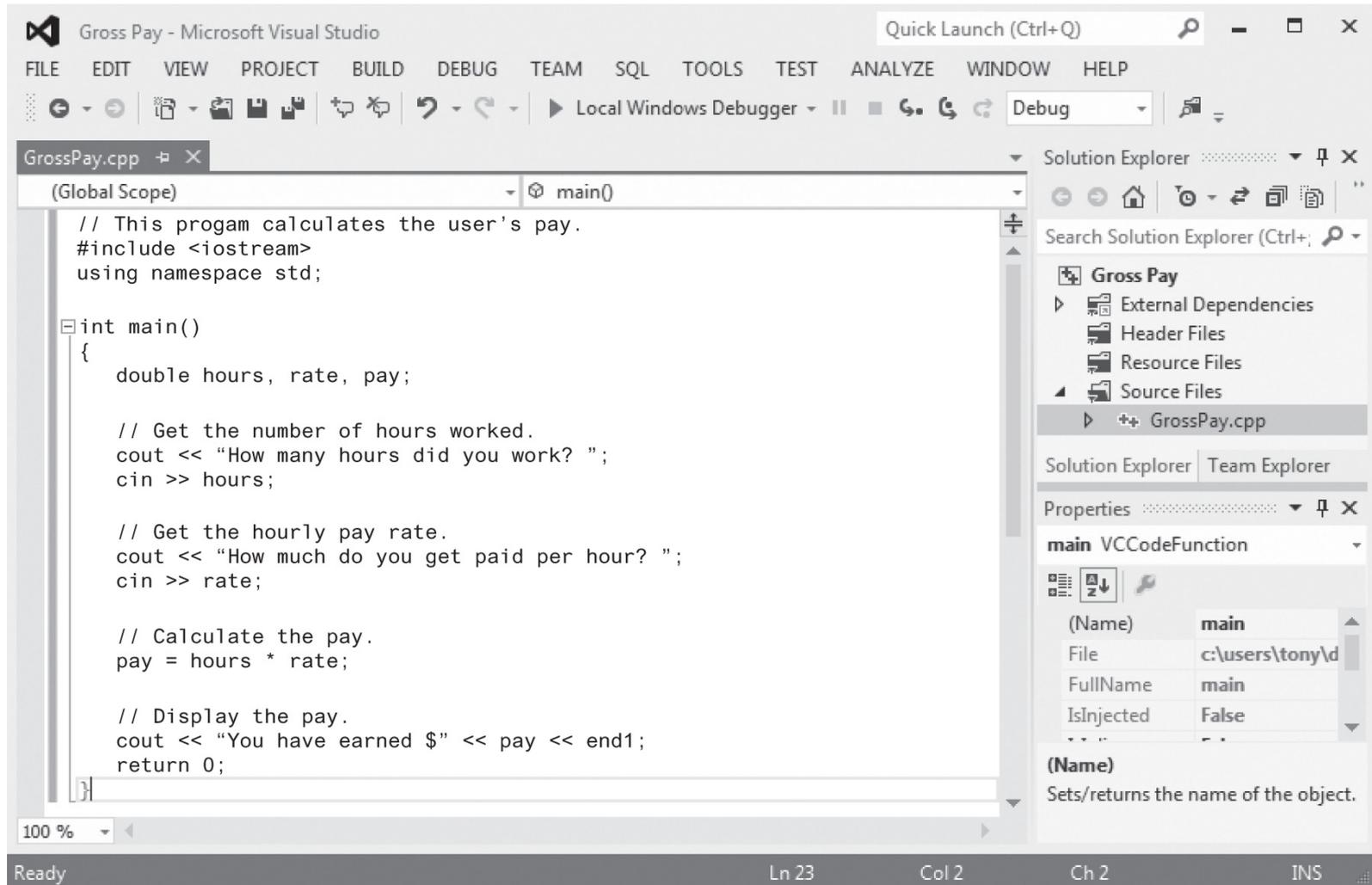
بيئات التطوير المتكاملة

Integrated Development Environments (IDEs)

- تجمع (تدمج) بيئات التطوير المتكاملة IDEs كل الأدوات (المعدات) اللازمة لكتابة write وترجمة compile وكشف خطأ debug برنامج ما في تطبيق برمجي واحد.
- أمثلة عن بعض بيئات التطوير:

Microsoft Visual C++, Turbo C++ Explorer,
CodeWarrior, etc.

Integrated Development Environments (IDEs)



1.4

مم يتكون البرنامج

What is a Program Made of?

مم يتكون البرنامج؟

What is a Program Made of?

- العناصر المشتركة (الشائعة) في لغات البرمجة هي:
 - الكلمات المفتاحية (المحجوزة) Key Words
 - المعارف (المحددات) المعرفة من قبل المستخدم - Programmer- Defined Identifiers
 - العوامل Operators
 - علامات الترقيم Punctuations
 - الصيغ (التراكيب) Syntax

Program 1-1

Program 1-1

```
1 // This program calculates the user's pay.
2 #include <iostream>
3 using namespace std;
4
5 int main()
6 {
7     double hours, rate, pay;
8
9     // Get the number of hours worked.
10    cout << "How many hours did you work? ";
11    cin >> hours;
12
13    // Get the hourly pay rate.
14    cout << "How much do you get paid per hour? ";
15    cin >> rate;
16
17    // Calculate the pay.
18    pay = hours * rate;
19
20    // Display the pay.
21    cout << "You have earned $" << pay << endl;
22    return 0;
23 }
```

الكلمات المفتاحية (المحجوزة) Key Words

- تعرف أيضا بالكلمات المحجوزة
- لها معنى خاص في لغة C++
- لا يمكن استخدامها لأي غرض آخر
- من الكلمات المفتاحية في البرنامج 1-1 السابق:

using, namespace, int, double, return

Key Words

```
1 // This program calculates the user's pay.
2 #include <iostream>
3 using namespace std;
4
5 int main()
6 {
7     double hours, rate, pay;
8
9     // Get the number of hours worked.
10    cout << "How many hours did you work? ";
11    cin >> hours;
12
13    // Get the hourly pay rate.
14    cout << "How much do you get paid per hour? ";
15    cin >> rate;
16
17    // Calculate the pay.
18    pay = hours * rate;
19
20    // Display the pay.
21    cout << "You have earned $" << pay << endl;
22    return 0;
23 }
```

المعرفات المحددة من قبل المستخدم

Programmer-Defined Identifiers

- أسماء توضع من قبل المبرمج
- هي ليست جزءا من اللغة C++
- تستخدم لتمثيل الأشياء المختلفة: المتغيرات (مواقع الذاكرة)، الدوال functions وغيرها.
- من هذه المعرفات المذكورة في البرنامج 1-1:
hours, rate, pay.

العوامل Operators

- تستخدم لتنفيذ عمليات على البيانات data
- هناك العديد من العوامل منها:
 - الرياضية مثل: +، -، *، /
 - الإسناد (التخصيص) مثل: =
- بعض العوامل في البرنامج 1-1:

<< >> = *

Operators

```
1 // This program calculates the user's pay.
2 #include <iostream>
3 using namespace std;
4
5 int main()
6 {
7     double hours, rate, pay;
8
9     // Get the number of hours worked.
10    cout << "How many hours did you work? ";
11    cin >> hours;
12
13    // Get the hourly pay rate.
14    cout << "How much do you get paid per hour? ";
15    cin >> rate;
16
17    // Calculate the pay.
18    pay = hours * rate;
19
20    // Display the pay.
21    cout << "You have earned $" << pay << endl;
22    return 0;
23 }
```

علامات الترقيم Punctuation

- هي محارف تسم (تشير إلى) نهاية عبارة ما أو تفصل العناصر في قائمة ما.
- من هذه العلامات في البرنامج 1-1:
; (الفاصلة المنقوطة) و , (الفاصلة)

Punctuation

```
1 // This program calculates the user's pay.
2 #include <iostream>
3 using namespace std;
4
5 int main()
6 {
7     double hours, rate, pay;
8
9     // Get the number of hours worked.
10    cout << "How many hours did you work? ";
11    cin >> hours;
12
13    // Get the hourly pay rate.
14    cout << "How much do you get paid per hour? ";
15    cin >> rate;
16
17    // Calculate the pay.
18    pay = hours * rate;
19
20    // Display the pay.
21    cout << "You have earned $" << pay << endl;
22    return 0;
23 }
```

الصيغ: Syntax

- هي القواعد التي يجب إتباعها عند كتابة برنامج ما.
- تتحكم باستخدام الكلمات المفتاحية والعوامل والرموز المحددة من قبل المستخدم وعلامات الترقيم

المتغيرات: Variables

- المتغير هو موقع تخزين مسمى في ذاكرة الحاسب يستخدم لتخزين (مسك) قطعة من البيانات.
- استخدمنا في البرنامج السابق ثلاثة متغيرات هي:
 - متغير **hours** استخدم لتخزين ساعات العمل المنفذة (المشغولة) hours worked
 - متغير **rate** استخدم لتخزين معدل الأجر pay rate
 - متغير **pay** استخدم لتخزين الأجر الإجمالي gross pay

تعريف المتغير: Variable Definition

- كي تنشئ متغيرا في برنامج ما يجب أن تكتب تعريفا للمتغير (يدعى أيضا التصريح عن المتغير)
- هذه عبارة من البرنامج 1-1 تعرف المتغيرات:

`double hours, rate, pay;`

Variable Definitions

- هناك العديد من الأنواع المختلفة للبيانات يمكن أن نتعلمها في هذا المقرر.
- يخزن (يحتوي، يمسك) المتغير نوعا محددًا من البيانات.
- يحدد تعريف المتغير نوع البيانات التي يمكن أن يخزنها المتغير واسم المتغير.

Variable Definitions

- مثلا السطر 7 من البرنامج 1-1 هو:

```
double hours, rate, pay;
```

- تشير كلمة double إلى أن المتغيرات hours و rate و pay يمكن أن تخزن (تمسك) أعدادا ذات فاصلة عائمة بدقة مضاعفة double-precision floating point numbers (Chapter 2)

1.5

الدخل والمعالجة والخرج

Input, Processing, and Output

Input, Processing, and Output

نموذجيا ينفذ البرنامج ثلاث خطوات:

- (١) يجمع بيانات الدخل من لوحة المفاتيح أو من ملفات على سواقات الأقراص.
- (٢) يعالج بيانات الدخل تلك.
- (٣) يظهر النتائج كخرج بحيث يرسلها إلى الشاشة أو يكتبها في ملف ما.

1.6

عملية البرمجة

The Programming Process

The Programming Process

١. حدد بوضوح ما الذي يجب أن ينفذه البرنامج.
٢. تصور أن البرنامج ينفذ على الحاسب.
٣. استخدم أدوات التصميم مثل المخطط الهرمي ومخططات التدفق أو الشيفرة المزيفة لإنشاء نموذج للبرنامج.
٤. اختبر النموذج بالنسبة للأخطاء المنطقية.
٥. اكتب الشيفرة وخرنها ثم قم بترجمتها باستخدام المترجم.
٦. صحح أية أخطاء موجودة أثناء الترجمة. كرر الخطوات 5 و6 عدة مرات حسب الحاجة.
٧. نفذ البرنامج بإدخال بيانات اختبار.
٨. صحح أية أخطاء موجودة أثناء تنفيذ البرنامج. كرر الخطوات 5 حتى 8 عدة مرات حسب الحاجة.
٩. صحح نتائج البرنامج.

1.7

البرمجة الإجرائية والبرمجة الشيئية (غرضية التوجه)

Procedural and Object-Oriented Programming

Procedural and Object-Oriented Programming

- البرمجة الإجرائية: ينصب التركيز على العملية (المعالجة). تكتب الإجراءات/الدوال كي تعالج البيانات.
- البرمجة الشيئية: تركز على الأشياء (الكائنات) والتي تحتوي على البيانات والوسائل التي تعالج البيانات. ترسل الرسائل إلى الكائنات كي تنفذ العمليات.

Copyright

