

C++ Programming: From Problem Analysis to Program Design, Fifth Edition

Chapter 1: An Overview of Computers and Programming Languages

Review on Problem Solving

مراجعة حول حل المشكلات

Solving Problems Using

حل المشكلات باستخدام
خرائط التدفق

Flowcharts

- A flowchart is a graphical representation of an algorithm. These flowcharts play a vital role in the programming of a problem and are quite helpful in understanding the logic of complicated and lengthy problems.

• مخطط التدفق هو تمثيل رسومي للخوارزمية.

- Once the flowchart is drawn, it becomes easy to write the program in any high level language. Often we see how flowcharts are helpful in explaining the program to others. Hence, it is correct to say that a flowchart is a must for the better documentation of a complex program.

Flowcharts Standard Symbols



Start or end of the program



Computational steps or processing function of a program



Input or output operation



Decision making and branching

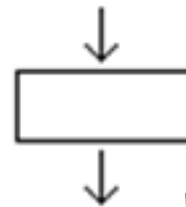
Flowchart Guidelines

إرشادات المخطط الانسيابي

In drawing a proper flowchart, all necessary requirements should be listed out in logical order. The flowchart should be clear, neat and easy to follow. The usual direction of the flow of a procedure or system is from left to right or top to bottom. Only one flow line should come out from a process symbol.



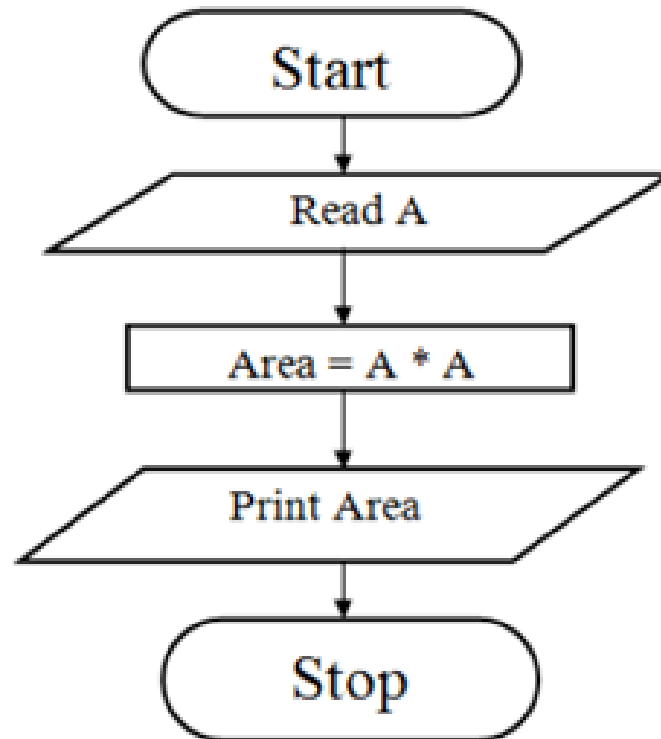
or



عند رسم مخطط انسيابي مناسب ،
يجب سرد جميع المتطلبات الضرورية
بترتيب منطقي. يجب أن يكون المخطط
الانسيابي واضحاً وأنيقاً وسهل المتابعة.
الاتجاه المعتاد لتدفق إجراء أو نظام
هو من اليسار إلى اليمين أو من أعلى إلى أسفل.
يجب أن يخرج خط تدفق واحد فقط من رمز العملية.

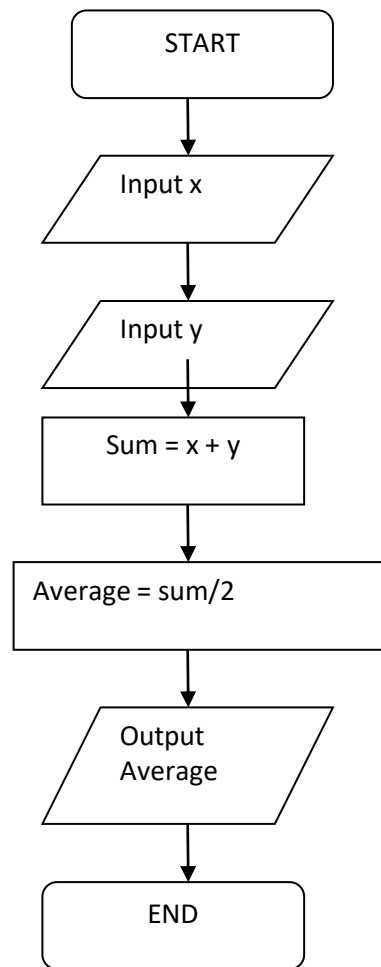
Example 1-1

Write an algorithm and draw the flowchart for finding the area of a square



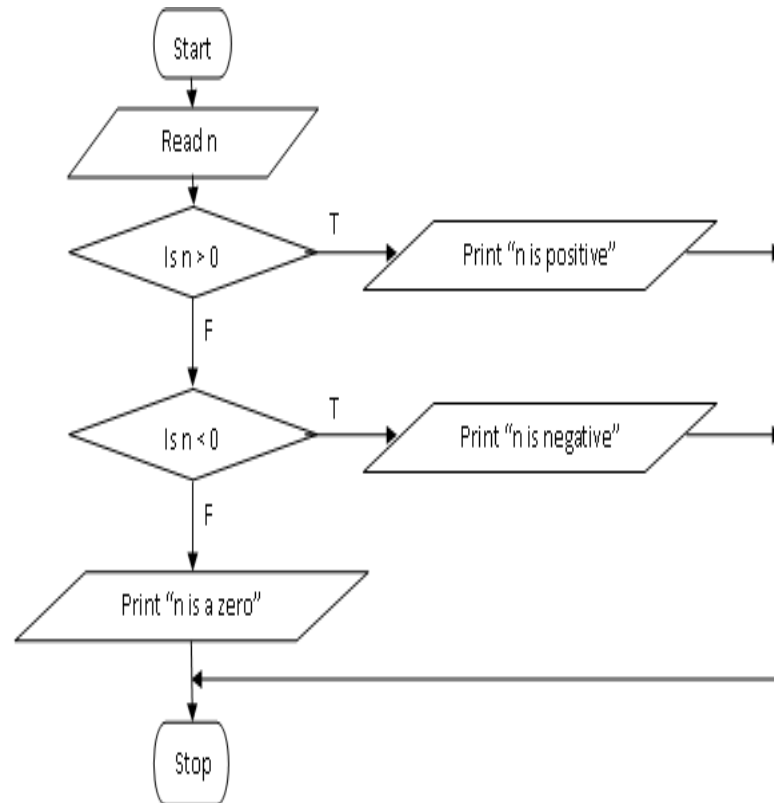
Example 1-2

Draw the
flowchart for
finding the
average of
two numbers



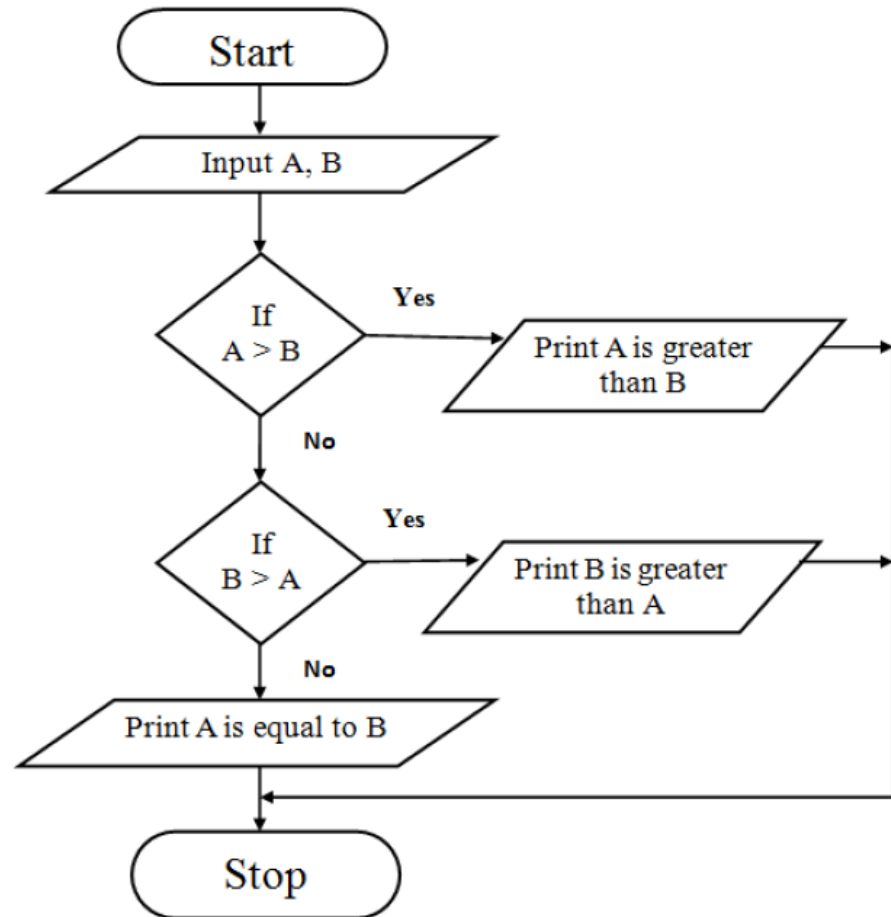
Example 1-3

Draw the flowchart to find if a given number is Negative, Positive, or Zero



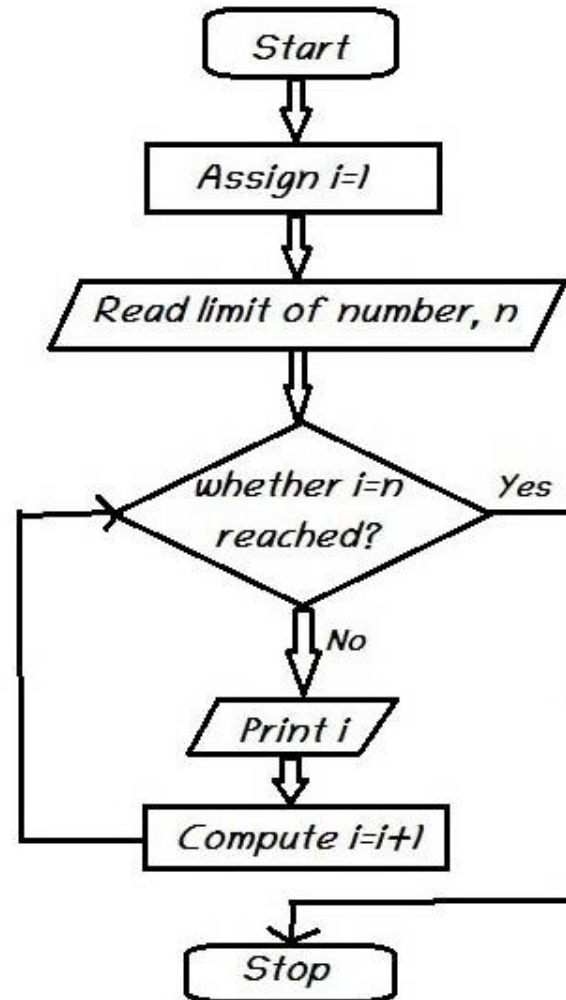
Example 1-4

Draw the flowchart to find the largest of two numbers



Example 1-5

Draw a flowchart for a program that prints the numbers from 1 to n



Example 1-6

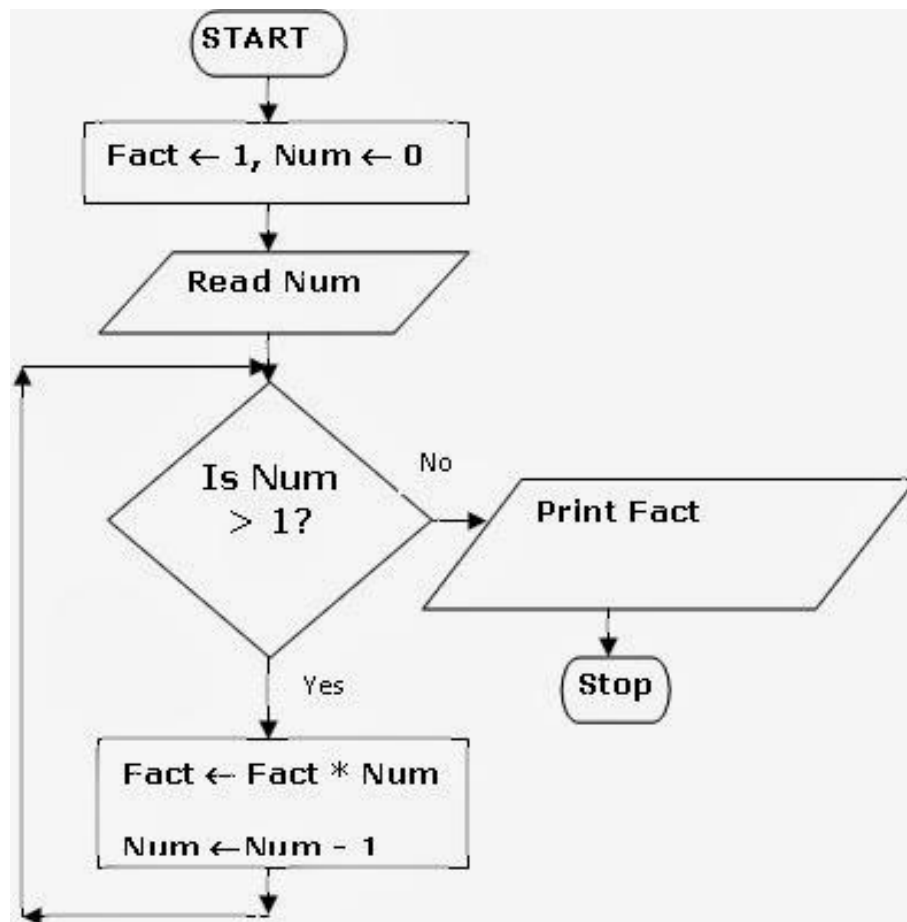
طريقة الحل :

-1 design

-2 trace

For a non-negative integer n , draw a flowchart for a program *that finds the factorial of n (written as $n!$)*

للعدد الصحيح غير السالب n ، ارسم مخططًا انسيابيًا
لبرنامج يجد عاملي n (مكتوبًا كـ $n!$)



Review on Fundamental Concepts in Programming

مراجعة المفاهيم الأساسية في البرمجة

Why Programming?

- Without software, the computer is useless
- Software developed with programming languages
 - C++ is a programming language
- C++ suited for a wide variety of programming tasks
- Before programming, it is useful to understand terminology and computer components

• بدون برنامج ، يكون الكمبيوتر عديم الفائدة

• C++ مناسبة لمجموعة متنوعة من مهام البرمجة

• قبل البرمجة ، من المفيد فهم المصطلحات ومكونات الكمبيوتر

Elements of a Computer System

عناصر نظام الكمبيوتر

- CPU
 - Main memory
 - Secondary storage
 - Input/Output devices
 - Software
- وحدة المعالجة المركزية
 - الذاكرة الرئيسية
 - التخزين الثانوي
 - أجهزة الإدخال / الإخراج
 - برمجة

Central Processing Unit and Main Memory

وحدة المعالجة المركزية والذاكرة الرئيسية

- Central processing unit • وحدة معالجة مركزية
 - Brain of the computer
 - Most expensive piece of hardware
 - Carries out arithmetic and logical operations
- دماغ الكمبيوتر
- أعلى قطعة من الأجهزة
- يقوم بعمليات حسابية ومنطقية

Central Processing Unit and Main Memory (cont'd.)

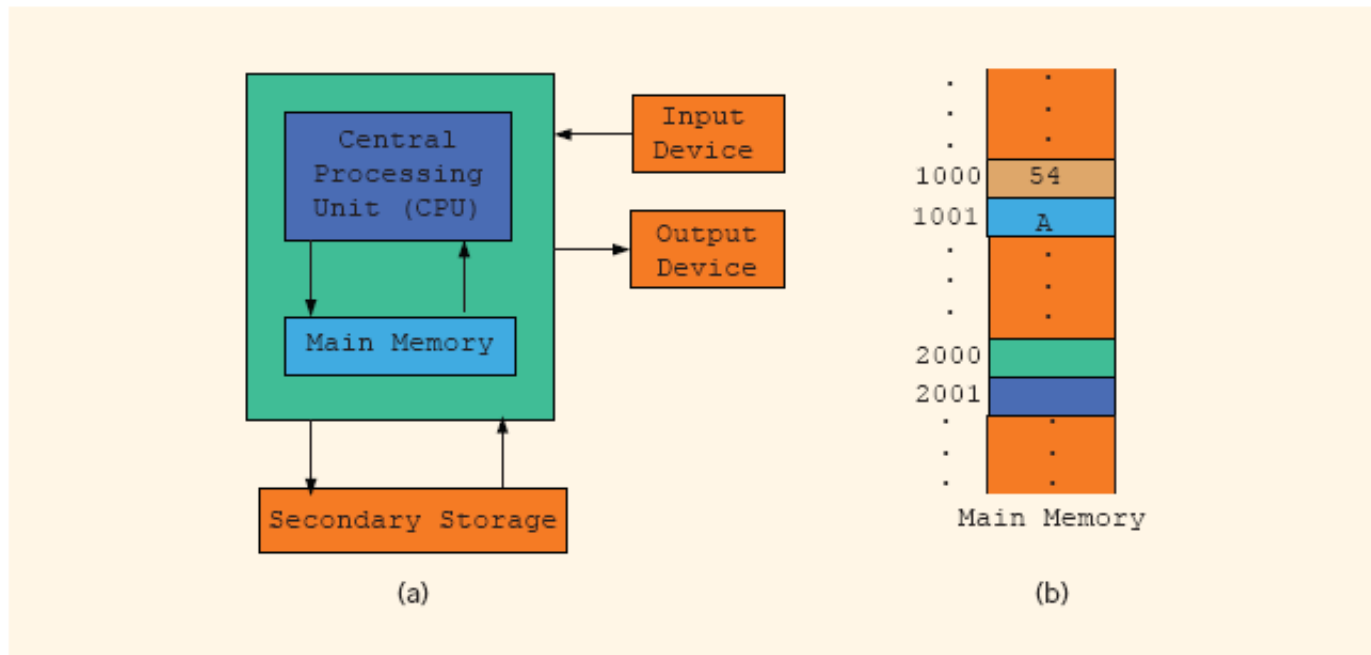


FIGURE 1-1 Hardware components of a computer and main memory

Central Processing Unit and Main Memory (cont'd.)

وحدة المعالجة المركزية والذاكرة الرئيسية (تابع)

مهم
Q
(T/F)

- Random access memory • ذاكرة الوصول العشوائي
- Directly connected to the CPU • متصل مباشرة بوحدة المعالجة المركزية
- All programs must be loaded into main memory before they can be executed • يجب تحميل كافة البرامج في الذاكرة الرئيسية قبل أن يتم تنفيذها
- All data must be brought into main memory before it can be manipulated • يجب إحضار جميع البيانات إلى الذاكرة الرئيسية قبل التمكن من معالجتها
- When computer power is turned off, everything in main memory is lost • عند إيقاف تشغيل الكمبيوتر ، يتم فقد كل شيء في الذاكرة الرئيسية

Secondary Storage

التخزين الثانوي

- Secondary storage: device that stores information permanently
- أمثلة على التخزين الثانوي: التخزين الثانوي: جهاز يخزن المعلومات بشكل دائم
 - Hard disks - الأقراص الصلبة
 - Flash drives - ذاكرة فلاش
 - Floppy disks - الأقراص المرنة
 - Zip disks - أقراص مضغوطة
 - CD-ROMs - أقراص مدمجة
 - Tapes - شرائط

Input/Output Devices

أجهزة الإدخال / الإخراج

- Input devices feed data and programs into computers
 - Keyboard - لوحة المفاتيح
 - Mouse - الفأر
 - Secondary storage - تخزين ثانوي
- Output devices display results
 - Monitor - شاشة
 - Printer - طابعة
 - Secondary storage - تخزين ثانوي

Software

• البرمجيات: البرامج التي تقوم بمهام محددة

- Software: programs that do specific tasks
- System programs take control of the computer, such as an operating system
- Application programs such as:

• تتحكم برامج النظام في الكمبيوتر، مثل نظام التشغيل

 - Word processors - معالجات النصوص
 - Spreadsheets - جداول البيانات
 - Games - ألعاب

The Language of a Computer

لغة الحاسوب

- Digital signals: sequences of 0s and 1s
• الإشارات الرقمية: متواليات من 0 و 1
- Machine language: language of a computer
• لغة الآلة: لغة الحاسوب
- Binary digit (bit):
 - The digit 0 or 1
 - رقم ثنائي (بت):
 - الرقم 0 أو 1
- Binary code:
 - A sequence of 0s and 1s
 - الكود الثنائي:
 - تسلسل من 0 و 1
- Byte:
 - A sequence of eight bits
 - البايت:
 - سلسلة من ثماني بتات

The Language of a Computer (cont'd.)

TABLE 1-1 Binary Units

Unit	Symbol	Bits/Bytes
Byte		8 bits
Kilobyte	KB	2^{10} bytes = 1024 bytes
Megabyte	MB	$1024 \text{ KB} = 2^{10} \text{ KB} = 2^{20} \text{ bytes} = 1,048,576 \text{ bytes}$
Gigabyte	GB	$1024 \text{ MB} = 2^{10} \text{ MB} = 2^{30} \text{ bytes} = 1,073,741,824 \text{ bytes}$
Terabyte	TB	$1024 \text{ GB} = 2^{10} \text{ GB} = 2^{40} \text{ bytes} = 1,099,511,627,776 \text{ bytes}$
Petabyte	PB	$1024 \text{ TB} = 2^{10} \text{ TB} = 2^{50} \text{ bytes} = 1,125,899,906,842,624 \text{ bytes}$
Exabyte	EB	$1024 \text{ PB} = 2^{10} \text{ PB} = 2^{60} \text{ bytes} = 1,152,921,504,606,846,976 \text{ bytes}$
Zettabyte	ZB	$1024 \text{ EB} = 2^{10} \text{ EB} = 2^{70} \text{ bytes} = 1,180,591,620,717,411,303,424 \text{ bytes}$

The Language of a Computer

(cont'd.) لغة الحاسوب (تابع)

- ASCII (American Standard Code for Information Interchange)
 - 128 characters
 - A is encoded as 1000001 (66th character)
 - 3 is encoded as 0110011

• ASCII (الكود القياسي الأمريكي لتبادل المعلومات)

- 128 حرفاً

- تم ترميز A كـ 1000001 (الحرف 66)

- 3 تم ترميزه على أنه 0110011

The Evolution of Programming Languages

تطور لغات البرمجة

- Early computers were programmed in machine language
- تمت برمجة أجهزة الكمبيوتر القديمة بلغة الآلة
- To calculate `wages = rates * hours` in machine language:

100100 010001 //Load

100110 010010 //Multiply

100010 010011 //Store

The Evolution of Programming Languages (cont'd.)

تطور لغات البرمجة (تابع)

- Assembly language instructions are mnemonic
- Assembler: translates a program written in assembly language into machine language

• تعليمات لغة التجميع هي ذاكري

TABLE 1-2 Examples of Instructions in Assembly Language and Machine Language

Assembly Language	Machine Language
LOAD	100100
STOR	100010
MULT	100110
ADD	100101
SUB	100011

• المجمع: يترجم البرنامج المكتوب بلغة التجميع إلى لغة الآلة

The Evolution of Programming Languages (cont'd.)

- Using assembly language instructions,
`wages = rates • hours` can be
written as:

```
LOAD  rate
```

```
MULT  hour
```

```
STOR  wages
```

The Evolution of Programming Languages (cont'd.)

تطور لغات البرمجة (تابع)

- High-level languages include Basic, FORTRAN, COBOL, Pascal, C, C++, C#, and Java
• تشمل اللغات عالية المستوى Basic و FORTRAN و COBOL و Pascal و C و C++ و C# و Java
- Compiler: translates a program written in a high-level language into machine language
• المترجم: يترجم البرنامج المكتوب بلغة عالية المستوى إلى لغة الآلة
- The equation $wages = rate \cdot hours$ can be written in C++ as:

```
wages = rate * hours;
```

The Problem Analysis–Coding–Execution Cycle (cont'd.)

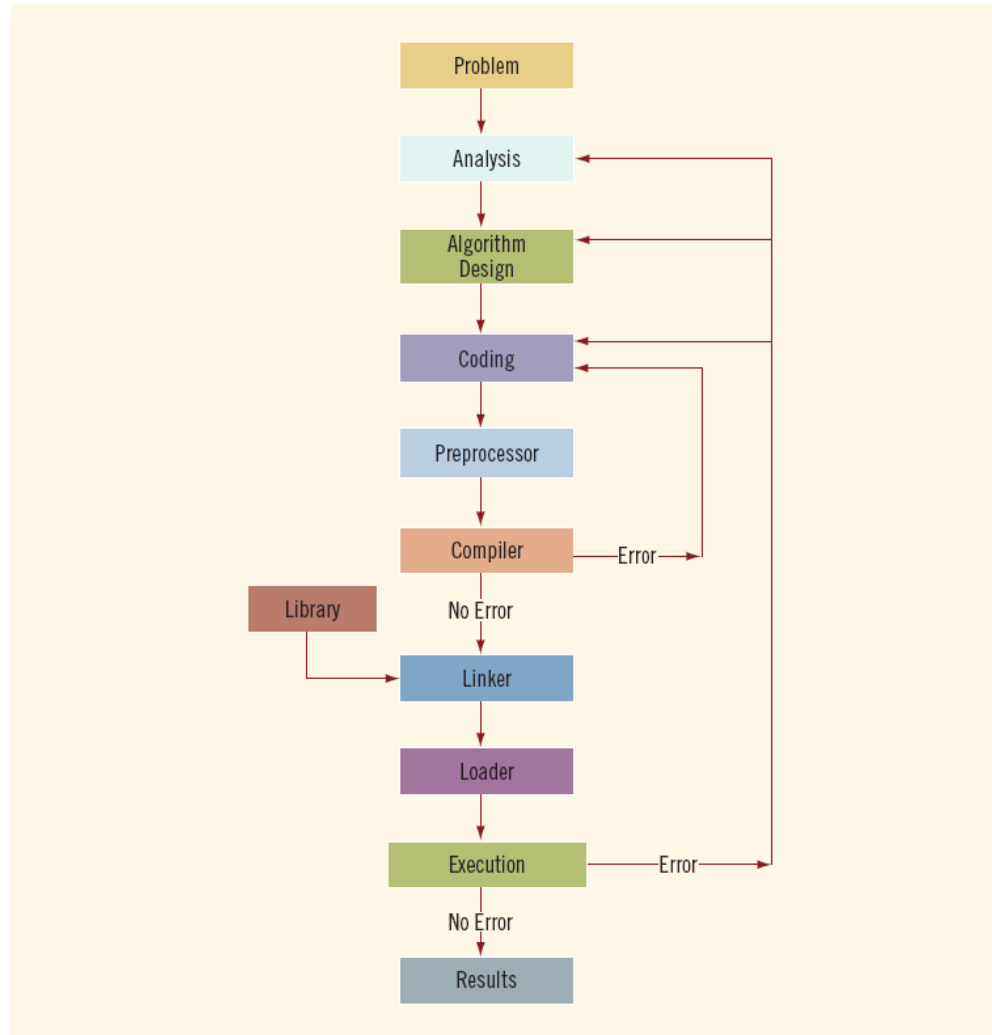


FIGURE 1-4 Problem analysis–coding–execution cycle

Programming with the Problem Analysis–Coding–Execution Cycle

البرمجة مع تحليل المشكلة - الترميز - دورة التنفيذ

- Programming is a process of problem solving
 - البرمجة هي عملية حل المشكلات
- One problem-solving technique: أسلوب واحد لحل المشكلات:
 - Analyze the problem - حل المشكلة
 - Outline the problem requirements - تحديد متطلبات المشكلة
 - Design steps (algorithm) to solve the problem
- Algorithm: الخوارزمية:
 - Step-by-step problem-solving process
 - Solution achieved in finite amount of time
 - خطوات التصميم (الخوارزمية) لحل المشكلة
 - عملية حل المشكلات خطوة بخطوة
 - تحقق الحل في مدة زمنية محدودة

The Problem Analysis–Coding–Execution Cycle (cont'd.)

دورة تحليل المشكلة - الترميز - التنفيذ (تابع)

- Thoroughly understand the problem فهم المشكلة بدقة
- Understand problem requirements فهم متطلبات المشكلة
 - Does program require user interaction?
- هل البرنامج يتطلب تفاعل المستخدم؟
 - Does program manipulate data?
- هل يتعامل البرنامج مع البيانات؟
 - What is the output?
- ما هو الناتج؟
- If the problem is complex, divide it into subproblems إذا كانت المشكلة معقدة ، قسّمها إلى مشاكل فرعية
 - Analyze each subproblem as above
- تحليل كل مشكلة فرعية على النحو الوارد أعلاه

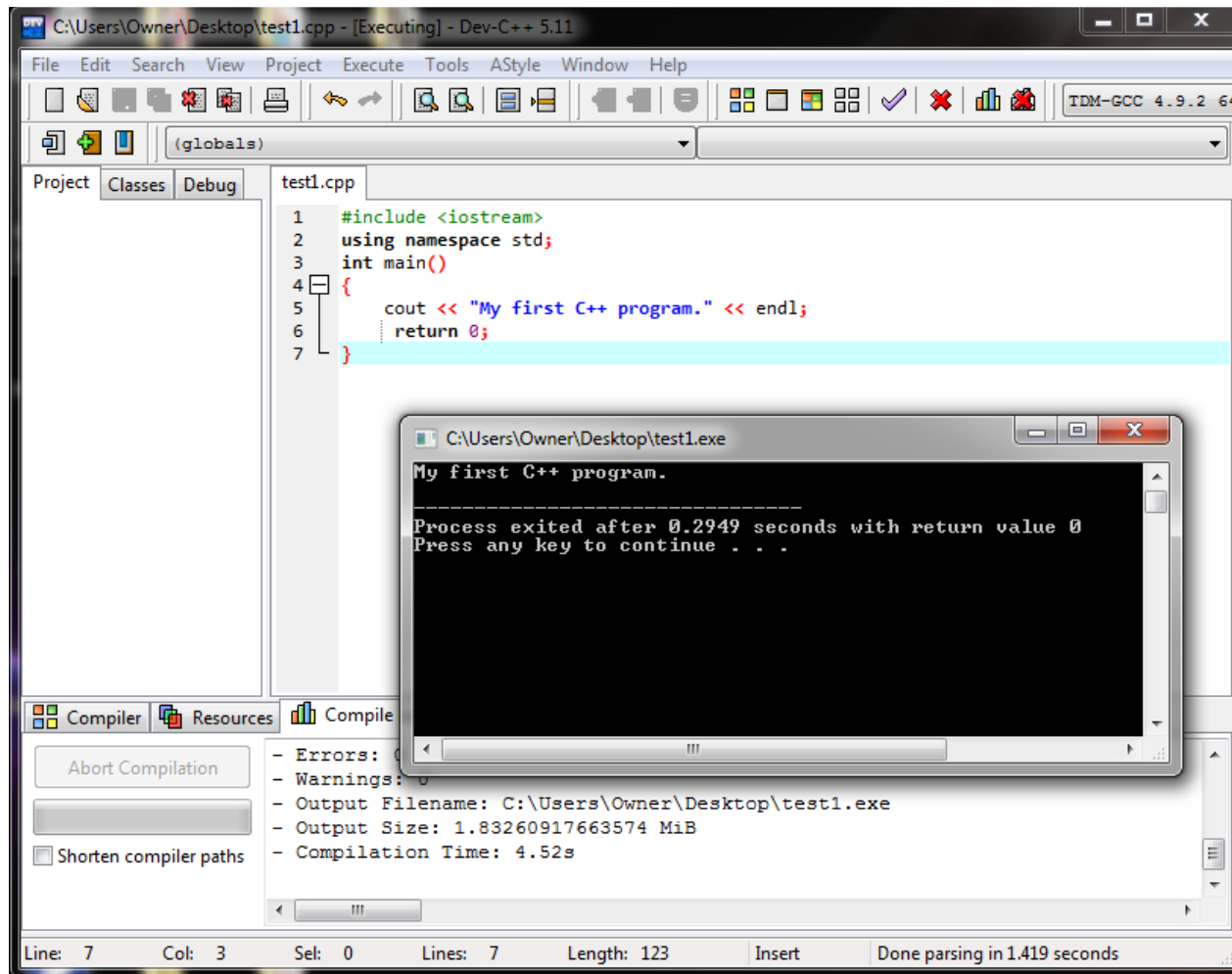
The Problem Analysis–Coding–Execution Cycle (cont'd.)

- If problem was broken into subproblems
 - Design algorithms for each subproblem
- Check the correctness of algorithm
 - Can test using sample data
 - Some mathematical analysis might be required

Processing a C++ Program

- To execute a C++ program:
 - Use an editor to create a source program in C++ (DEV C++ for example)
 - Preprocessor directives begin with # and are processed by a the preprocessor
 - Use the compiler to:
 - Check that the program obeys the rules
 - Translate into machine language (object program)

Processing a C++ Program



Processing a C++ Program (cont'd.)

- To execute a C++ program (cont'd.):
 - Linker:
 - Combines object program with other programs provided by the SDK to create executable code
 - Loader:
 - Loads executable program into main memory
 - The last step is to execute the program

The Problem Analysis–Coding–Execution Cycle (cont'd.)

- Once the algorithm is designed and correctness verified
 - Write the equivalent code in high-level language
- Enter the program using text editor

The Problem Analysis–Coding–Execution Cycle (cont'd.)

- Run code through compiler
- If compiler generates errors
 - Look at code and remove errors
 - Run code again through compiler
- If there are no syntax errors
 - Compiler generates equivalent machine code
- Linker links machine code with system resources

The Problem Analysis–Coding–Execution Cycle (cont'd.)

- Once compiled and linked, loader can place program into main memory for execution
- The final step is to execute the program
- Compiler guarantees that the program follows the rules of the language
 - Does not guarantee that the program will run correctly

Problem Challenges

Solving Problems Using Pseudo Code

حل المشكلات باستخدام الكود الزائف

Example 1-7

- Design an algorithm to find the perimeter and area of a rectangle
• تصميم خوارزمية لإيجاد محيط ومساحة المستطيل
- The perimeter and area of the rectangle are given by the following formulas:
• يُعطى محيط المستطيل ومساحته بالصيغ التالية:

```
perimeter = 2 * (length + width)
area = length * width
```

Example 1-7 (cont'd.)

- Algorithm:
 - Get length of the rectangle - احصل على طول المستطيل
 - Get width of the rectangle - احصل على عرض المستطيل
 - Find the perimeter using the following equation:
- أوجد المحيط باستخدام المعادلة التالية:
$$\text{perimeter} = 2 * (\text{length} + \text{width})$$
 - Find the area using the following equation:
- أوجد المنطقة باستخدام المعادلة التالية:
$$\text{area} = \text{length} * \text{width}$$

Example 1-8

- Every salesperson has a base salary
 - Salesperson receives \$10 bonus at the end of the month for each year worked if he or she has been with the store for five or less years
 - The bonus is \$20 for each year that he or she has worked there if over 5 years
- كل مندوب مبيعات له راتب أساسي
- يتلقى مندوب المبيعات مكافأة قدرها ١٠ دولارات أمريكية في نهاية الشهر عن كل عام عمل إذا كان يعمل في المتجر لمدة خمس سنوات أو أقل
- المكافأة هي ٢٠ دولارًا أمريكيًا عن كل عام عمل فيه هو أو هي هناك إذا كان أكثر من ٥ سنوات

Example 1-8 (cont'd.)

- المكافآت الإضافية كما يلي: Additional bonuses are as follows:
 - If total sales for the month are \$5,000-\$10,000, he or she receives a 3% commission on the sale
- إذا كان إجمالي المبيعات للشهر يتراوح بين ٥٠٠٠ و ١٠٠٠٠ دولار، فإنه يتلقى عمولة بنسبة ٣٪ على البيع
 - If total sales for the month are at least \$10,000, he or she receives a 6% commission on the sale
- إذا كان إجمالي المبيعات للشهر لا يقل عن ١٠٠٠٠ دولار، فإنه يتلقى عمولة بنسبة ٦٪ على البيع

Example 1-8 (cont'd.)

- **Get** `baseSalary`
- **Get** `noOfServiceYears`
- **Calculate** bonus using the following formula:

```
if (noOfServiceYears is less than or equal to  
five)
```

```
    bonus = 10 * noOfServiceYears
```

```
otherwise
```

```
    bonus = 20 * noOfServiceYears
```

- **Get** `totalSales`

Example 1-8 (cont'd.)

- **Calculate** `additionalBonus` as follows:

```
if (totalSale is less than 5000)
    additionalBonus = 0
otherwise
    if (totalSale is greater than or equal to
        5000 and totalSale is less than 10000)
        additionalBonus = totalSale * (0.03)
    otherwise
        additionalBonus = totalSale * (0.06)
```

Example 1-8 (cont'd.)

- **Calculate** `payCheck` using the equation

$$\text{payCheck} = \text{baseSalary} + \text{bonus} \\ + \text{additionalBonus}$$

Example 1-9

- 10 students in a class • ١٠ طلاب في الفصل
- Each student has taken five tests and each test is worth 100 points • أجرى كل طالب خمسة اختبارات وكل اختبار يساوي ١٠٠ نقطة
- Design an algorithm to calculate the grade for each student as well as the class average
 - تصميم خوارزمية لحساب الدرجة لكل طالب وكذلك متوسط الفصل
 - Design an algorithm to find the average test score
 - تصميم خوارزمية لإيجاد متوسط درجات الاختبار
 - Design an algorithm to determine the grade
 - تصميم خوارزمية لتحديد الدرجة
- Data consists of students' names and their test scores • تتكون البيانات من أسماء الطلاب ودرجاتهم في الاختبار

Example 1-9 (cont'd.)

- Algorithm to determine the average test score:
 - Get the five test scores
 - Add the five test scores
 - Suppose `sum` stands for the `sum` of the test scores
 - Suppose `average` stands for the average test score:
 - `average = sum / 5;`

• خوارزمية لتحديد متوسط درجات الاختبار:

Example 1-9 (cont'd.)

- Algorithm to determine the grade:

```
if average is greater than or equal to 90
    grade = A
otherwise
    if average is greater than or equal to 80 and less than 90
        grade = B
    otherwise
        if average is greater than or equal to 70 and less than 80
            grade = C
        otherwise
            if average is greater than or equal to 60 and less than 70
                grade = D
            otherwise
                grade = F
```


Example 1-9 (cont'd.)

- Main algorithm is as follows:
 - `totalAverage = 0;`
 - *Repeat* the following for each student:
 - Get student's name
 - Use the algorithm to find the average test score
 - Use the algorithm to find the grade
 - Update `totalAverage` by adding current student's average test score
 - Determine the class average as follows:
 - `classAverage = totalAverage / 10`
- كرر ما يلي لكل طالب:
- تحديد معدل الفصل على النحو التالي:

Programming Methodologies

منهجيات البرمجة

- Two popular approaches to programming design
 - Structured - منظم
 - Object Oriented Programming - البرمجة الشيئية

Structured Programming

برمجة منظمة

- Structured design: التصميم الهيكلي:
 - Dividing a problem into smaller subproblems
- Structured programming: برمجة منظمة:
 - Implementing a structured design
- The structured design approach is also called:
 - Top-down (or bottom-up) design
 - Stepwise refinement
 - Modular programming

- تقسيم المشكلة إلى مشاكل فرعية أصغر

•

- تنفيذ تصميم منظم

• يسمى نهج التصميم المنظم أيضًا:

- تصميم من أعلى إلى أسفل (أو من أسفل إلى أعلى)

- صقل تدريجي

- البرمجة المعيارية

Object-Oriented Programming

البرمجة الشيئية

- Identify components called objects
• تحديد المكونات التي تسمى الكائنات
- Specify relevant data and possible operations to be performed on that data
• تحديد البيانات ذات الصلة والعمليات الممكنة التي يتعين القيام بها على تلك البيانات
- Each object consists of data and operations on that data
• كل كائن يتكون من البيانات والعمليات على تلك البيانات
- An object combines data and operations on the data into a single unit
• كائن يجمع البيانات والعمليات على البيانات في وحدة واحدة

Object-Oriented Programming (cont'd.)

- Learn how to combine data and operations on the data into a single unit called an object
- C++ is designed to implement OOP
- OOP is used with structured design