

Introduction to Programming PYTHON

Chapter 1 - Overview

Presenter: **Dr. Nguyen Dinh Long**

Email: dinhlonghcmut@gmail.com

Google-site: <https://sites.google.com/view/long-dinh-nguyen>

Oct. 2022

Programming

Hamilton was a self-taught programmer, working in the US in the 1960's. Owing to the success of her previous work, Hamilton was the first programmer to be hired for the Apollo project. She became the Director of Software Engineering at the MIT Instrumentation lab. Her lab developed the on-board flight software for NASA's Apollo space project, which took humankind to the moon.

The achievement was a monumental task at a time when computer technology was in its infancy: The astronauts had access to only 72 kilobytes of computer memory (a 256-gigabyte cell phone today carries almost a million times more storage space). Programmers had to use paper punch cards to feed information into room-sized computers with no screen interface.

Margaret Hamilton, NASA's lead software engineer for the Apollo, stands next to the code she wrote by hand that took humanity to the moon in 1969.



Programming

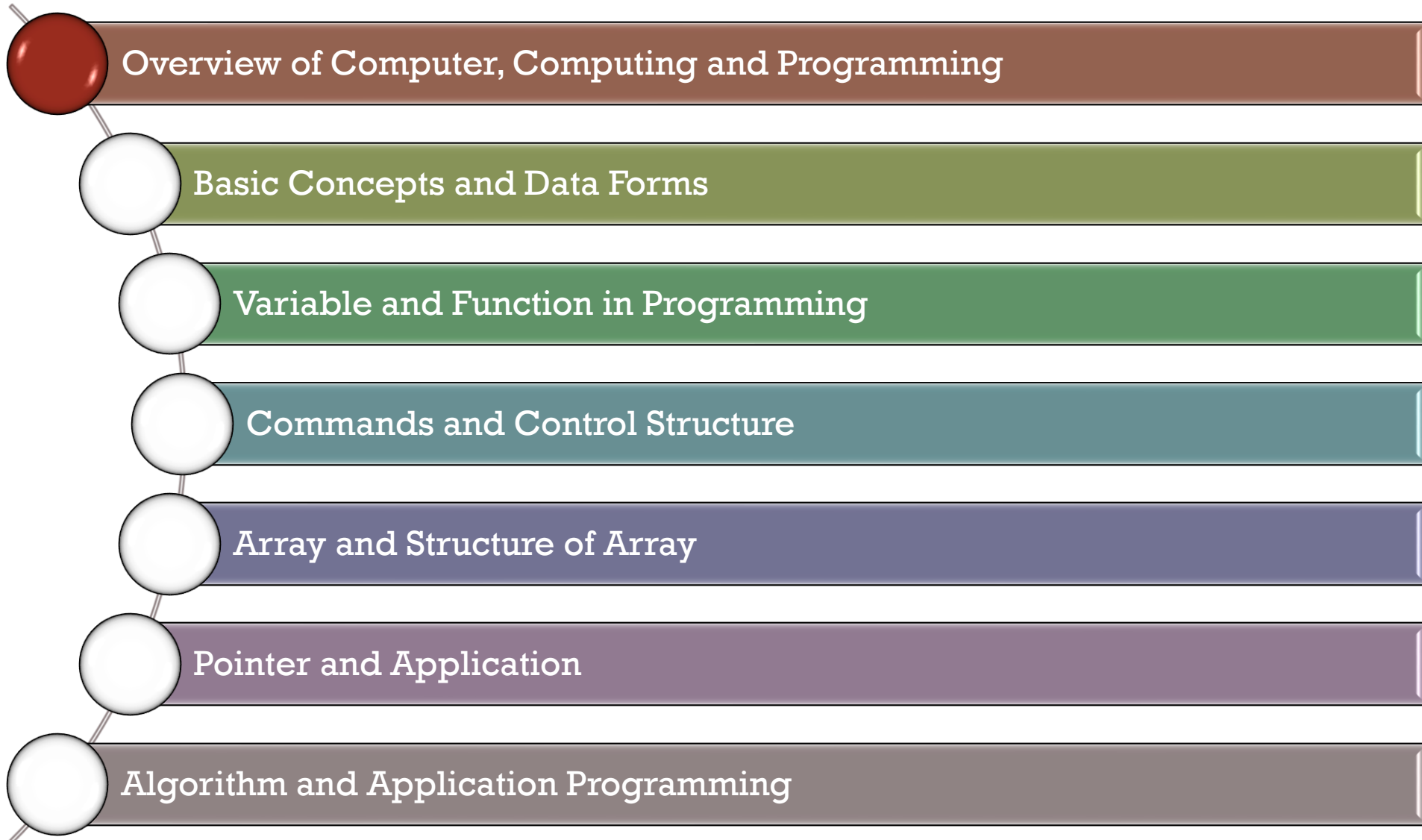
Margaret Hamilton, NASA's lead software engineer for the Apollo, stands next to the code she wrote by hand that took humanity to the moon in 1969.



In 2016, President Barack Obama awarded Hamilton the Medal of Freedom, noting that “her example speaks of the American spirit of discovery.” Official White House photo by Lawrence Jackson



Outline



References

Main:

- Maurizio Gabbrielli and Simone Martini, 2010. *Programming Languages: Principles and Paradigms*, Springer.
- Cao Hoàng Trữ, 2004. *Ngôn ngữ lập trình- Các nguyên lý và mô hình*, Nhà xuất bản Đại học Quốc gia Tp. Hồ Chí Minh

More:

- Robert W. Sebesta, 2008. *Concepts of Programming Languages*, 8th edition, Addison Wesley
- Kenneth C.Louden, 2003. *Programming Languages – Principles and Practice*, First edition, Thomson Brooks/Cole.

- Slides here are collected and modified from several sources in Universities and Internet.

Content of Chapter 1

1. Architecture of a computer
2. Programming languages
3. Program = objects + operations
4. Applications of Programming

What is a computer?

- ❑ Máy tính là thiết bị có khả năng thực hiện tính toán và ra quyết định.
- ❑ Máy tính hoạt động dựa trên nguyên lý là chạy các chương trình đã được định sẵn.
- ❑ Hiện nay, máy tính có thể được “nhúng” vào các thiết bị dân dụng như xe hơi, đồng hồ, điện thoại, máy tính phổ thông, v.v.



What is a computer?

Hardware



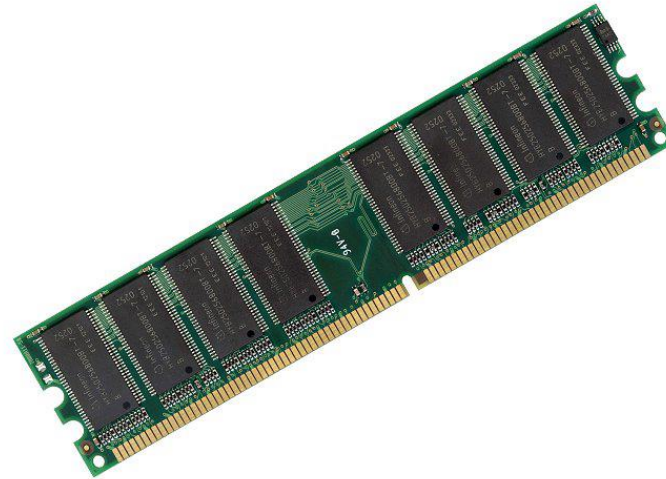
Software



What is a computer? - Hardware

□ Gồm các bộ phận dùng để lắp ráp thành bộ máy tính như :

- • Hộp máy (case),
- • Bo mạch chủ (mainboard, motherboard),
- • Bàn phím (keyboard),
- • Màn hình (display),
- • Chuột (mouse),
- • Thanh bộ nhớ (memory),
- • Ổ đĩa cứng HDD (Hard Disk Drive),
- • Bộ cấp nguồn (power supply),
- • Dây nối (cables),
- •



What is a computer? - Hardware



What is a computer? - Software

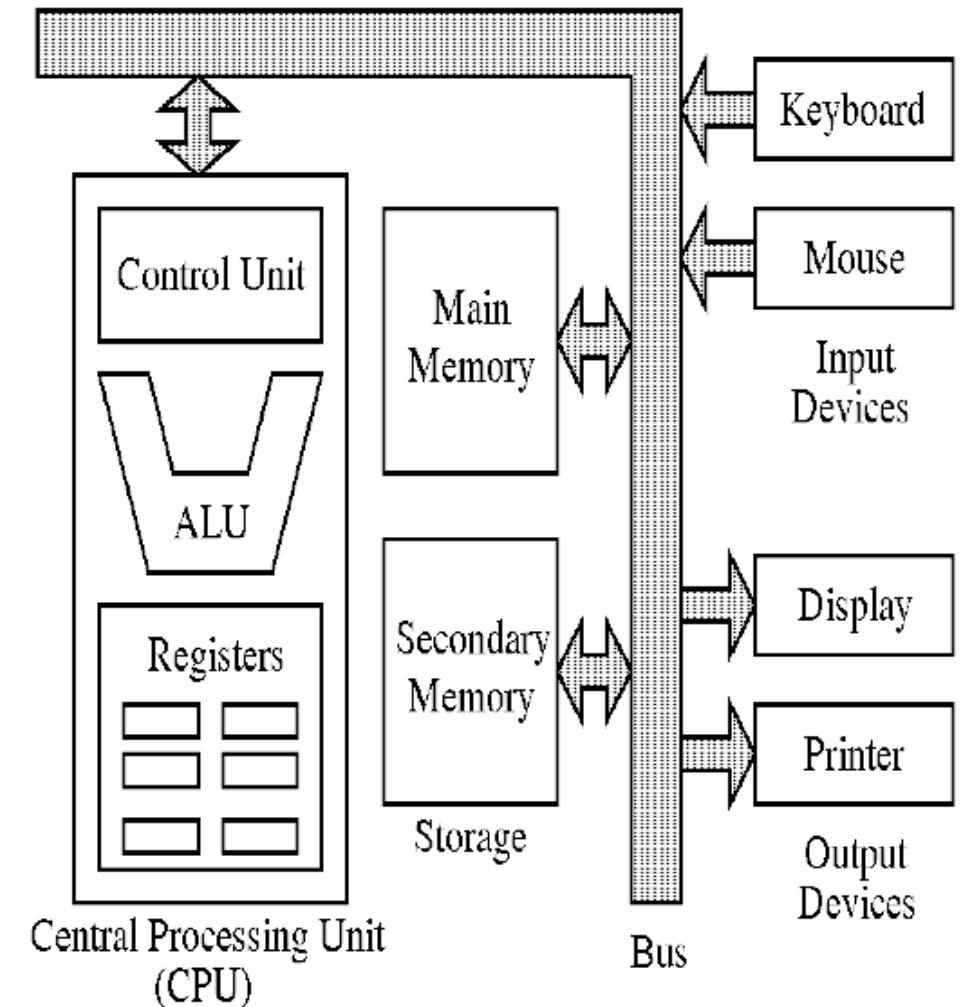
- ❑ Gồm các chương trình cài đặt sẵn giúp vận hành máy tính theo nhiều mục đích khác nhau.
- Hệ điều hành (operating system) : cho phép quản lý và khai thác tất cả phần cứng có trong máy tính.
- Công cụ lập trình (programming tools) : cho phép người sử dụng tạo ra thêm phần mềm mới trên máy, mở rộng phạm vi ứng dụng của máy tính.
- Phần mềm ứng dụng (applications) : cho phép khai thác máy tính theo mục đích cụ thể.

Examples of Operating System



Computer Operation

- ❑ Để điều khiển các thiết bị phần cứng như hình bên, chúng ta cần chương trình máy tính.
- ❑ Để tạo ra **chương trình máy tính**, chúng ta cần **ngôn ngữ lập trình**.



Programming Language

❑ Ngôn ngữ lập trình (Programming language)

- Là một ngôn ngữ hình thức (formal language), khác với ngôn ngữ tự nhiên (natural language), ví dụ ngôn ngữ lập trình C hoặc Python sẽ được học trong môn học này.
- Mục đích: cho phép con người (lập trình viên) tạo ra chương trình máy tính.

❑ Ba cấp độ ngôn ngữ lập trình :

1. Ngôn ngữ máy (machine language).
2. Ngôn ngữ hợp ngữ (assembly language).
3. Ngôn ngữ cấp cao (high-level language).

Programming Language

❑ Ngôn ngữ máy

- Dạng thức là số, đặc tả các lệnh của máy tính.
- Mỗi kiểu CPU có tập lệnh riêng.
- Ngày nay, rất khó có thể ra lệnh trực tiếp kiểu này cho các chương trình thực tế và lớn. Chuỗi lệnh số sẽ được sinh ra tự động từ bộ chuyển ngữ (compiler) từ ngôn ngữ cấp cao sang.

❑ Ví dụ: +1300042774, +1400593419, +1200274027

Programming Language

❑ Ngôn ngữ Assembly (hợp ngữ)

- Sử dụng các từ viết tắt trong tiếng Anh để biểu diễn các tác vụ tính toán cơ bản
- Bộ chuyển ngữ (sang mã máy) cho ngôn ngữ này là “Assembler”.
- Ngày nay, ngôn ngữ này vẫn còn được sử dụng, và thường kết hợp với ngôn ngữ cấp cao. Nó được dùng để tối ưu một số khối xử lý trong toàn bộ chương trình.

❑ Ví dụ:

LOAD BASEPAY

ADD OVERPAY

STORE GROSSPAY

Programming Language

❑ Ngôn ngữ C:

- Ngôn ngữ C/C++
- Hay ngắn gọn: C

❑ Lịch sử

- Tiến hóa từ ngôn ngữ lập trình trước đó là “B” và “BCPL”
- “Traditional C” là một tiến hóa của C vào khoảng gần 1970
- Có nhiều biến thể của C tồn tại và chúng không tương thích nhau, i.e., bộ chuyển ngữ không hiểu mã nguồn C được viết cho bộ chuyển ngữ khác.
- “Standard C” xuất hiện vào khoảng 1989, cập lại vào khoảng 1999.

❑ Ứng dụng quan trọng

- Được dùng để phát triển các hệ điều hành nổi tiếng và hiện đại như UNIX và Linux
- Được dùng để phát triển các chương trình chạy trên các thiết bị nhúng, như, nhúng vào xe hơi, máy móc y tế, etc.

Programming Language

Đẹp dễ tốt hơn xấu xí
Minh bạch tốt hơn ngầm định
Đơn giản tốt hơn phức tạp
Phức tạp tốt hơn rắc rối
Tính dễ đọc rất quan trọng.
(*dạng thơ Haiku*)

❑ Ngôn ngữ Python:

- Ngôn ngữ Python
- Hay ngắn gọn: Python

❑ Lịch sử

- Python đã được Guido van Rossum thai nghén vào những năm 1980.
- Python là một ngôn ngữ lập trình bậc cao cho các mục đích lập trình đa năng, do Guido van Rossum tạo ra và lần đầu ra mắt vào năm 1991.
- Van Rossum đã tự mình gánh vác trách nhiệm cho dự án, với vai trò là nhà phát triển chính, cho đến ngày 12 tháng Bảy năm 2018, khi thông báo rằng ông sẽ rời bỏ trách nhiệm của ông và cả danh hiệu "*Nhà độc tài nhân từ cho cuộc sống*"
- Python được phát triển để chạy trên nền Unix. Python dần mở rộng sang mọi hệ điều hành từ MS-DOS đến Mac OS, OS/2, Windows, Linux và các hệ điều hành khác thuộc họ Unix.
- Python luôn được xếp hạng vào những ngôn ngữ lập trình phổ biến nhất. Từng được chọn là Ngôn ngữ Lập trình của Năm ("có đánh giá tăng cao nhất trong năm") vào các năm 2007, 2010, 2018, và 2020.

❑ Ứng dụng quan trọng

- Sử dụng cho các tổ chức: Wikipedia, Google, Yahoo!, CERN, NASA, Facebook, Amazon, Instagram, Spotify
- Được dùng để ứng dụng web, tính toán khoa học, dự án trí tuệ nhân tạo, thư viện trò chơi, sản phẩm phần mềm, hệ điều hành sử dụng Python.

Programming Concepts

❑ Quan điểm:

- Chương trình là sự thể hiện giải thuật của bài toán bằng ngôn ngữ lập trình.
- Do đó, chương trình là tập hợp các lệnh và dữ liệu được viết đúng cú pháp của ngôn ngữ lập trình.
- Trình tự viết lệnh, sự kết hợp lệnh để tác động lên dữ liệu phải tuân thủ theo giải thuật của bài toán.

❑ Do đó, hai yếu tố quan trọng tạo thành chương trình:

1. **Dữ liệu và cấu trúc để tổ chức dữ liệu.**
2. **Giải thuật xử lý:** cách giải bài toán, cách biến đổi dữ liệu từ dữ liệu ban đầu, dần dần đi đến kết quả.

Programming Concepts - Data

❑ Kiểu dữ liệu cơ bản (ví dụ có ngôn ngữ C)

- Kiểu số
 - Số nguyên: int
 - Số thực chấm động: float, double
- Kiểu ký tự / chuỗi : char, dãy char
- Kiểu void: void
- Kiểu luận lý: bool (bổ sung bởi C++)
- Kiểu enum
- Kiểu dữ liệu do người lập trình định nghĩa: cấu trúc (struct), hợp nhất (union)
- Kiểu dẫn xuất: dãy (array), con trỏ (pointer)

Programming Concepts - Algorithm

❑ Phần đầu của giải thuật

- Tên giải thuật
- Thông số và kiểu của nó
- Mục đích của giải thuật
- Điều kiện cần thỏa trước khi giải thuật thực thi
- Điều kiện cần thỏa sau khi giải thuật thực thi
- Giá trị trả về

❑ Phần thân của giải thuật

- Các phát biểu
- Các biến
- Chú thích
- Các cấu trúc điều khiển

Phần đầu của giải thuật

Phần thân của giải
thuật

📌 Các cách khác nhau dùng để mô tả giải thuật

- Pseudocode = English/Vietnamese + Code
- Cú pháp không cần chặt chẽ
Để hỗ trợ việc dễ đọc
- Lệnh điều khiển thực thi
1. Tuần tự
 2. Rẽ nhánh
 3. Lặp



Programming Concepts - Algorithm

- Ví dụ
 - Thuật toán giải PT bậc nhất: $ax + b = 0$
(a, b là các số thực).

Đầu vào: a, b thuộc R

Đầu ra: nghiệm phương trình $ax + b = 0$

- Nếu $a = 0$
 - $b = 0$ thì phương trình có nghiệm bất kì.
 - $b \neq 0$ thì phương trình vô nghiệm.
- Nếu $a \neq 0$
 - Phương trình có nghiệm duy nhất $x = -b/a$



Programming Concepts - Algorithm

❑ Tính chất của thuật toán:

- **Tính chính xác:** quá trình tính toán hay các thao tác máy tính thực hiện là chính xác.
- **Tính rõ ràng:** các câu lệnh minh bạch được sắp xếp theo thứ tự nhất định.
- **Tính khách quan:** được viết bởi nhiều người trên máy tính nhưng kết quả phải như nhau.
- **Tính phổ dụng:** có thể áp dụng cho một lớp các bài toán có đầu vào tương tự nhau.
- **Tính kết thúc:** hữu hạn các bước tính toán.



Programming Concepts - Programming code

❑ Mục tiêu:

- Mã được viết bởi ngôn ngữ lập trình như C, C++, Python, Java, ...
- Chỉ dùng để trao đổi giữa những người có chuyên môn với nhau.
- Có thể chuyển ngữ dễ dàng và có chương trình thực thi ngay.
- Không nên dùng mã này để trình bày cho các đối tượng người dùng không chuyên nghiệp.



Programming Design

❑ Soạn thảo mã nguồn

- Đối tượng thực hiện: Người lập trình
- Công cụ cần đến:
 - Trình soạn thảo đơn giản như NOTEPAD. Nhưng ít khi được dùng
 - Trình soạn thảo tích hợp trong IDE (Integrated Development Environment)
- Đầu ra: tập tin mã nguồn, dạng văn bản đọc được

❑ Tiền xử lý (Preprocess)

- Đối tượng thực hiện: Preprocessor (bộ tiền xử lý), thuộc IDE
- Công việc thực hiện: tiền xử lý chương trình, như, thay các “macro” trong bởi phần định nghĩa của nó, chèn các tập tin khai báo thư viện (v.d., stdio.h)

❑ Biên dịch (Compile)

- Đối tượng thực hiện: Compiler (bộ biên dịch), thuộc IDE
- Công việc thực hiện: Chuyển mã, từ mã C sang mã đối tượng, cho từng tập tin mã nguồn

❑ Liên kết (Link)

- Đối tượng thực hiện: Linker (bộ liên kết), thuộc IDE
- Công việc thực hiện: Liên kết các tập tin mã đối tượng và thư viện của NNLT để tạo chương trình thực thi
- Đầu ra: tập tin thực thi (*.exe)

PROGRAMMING DESIGN

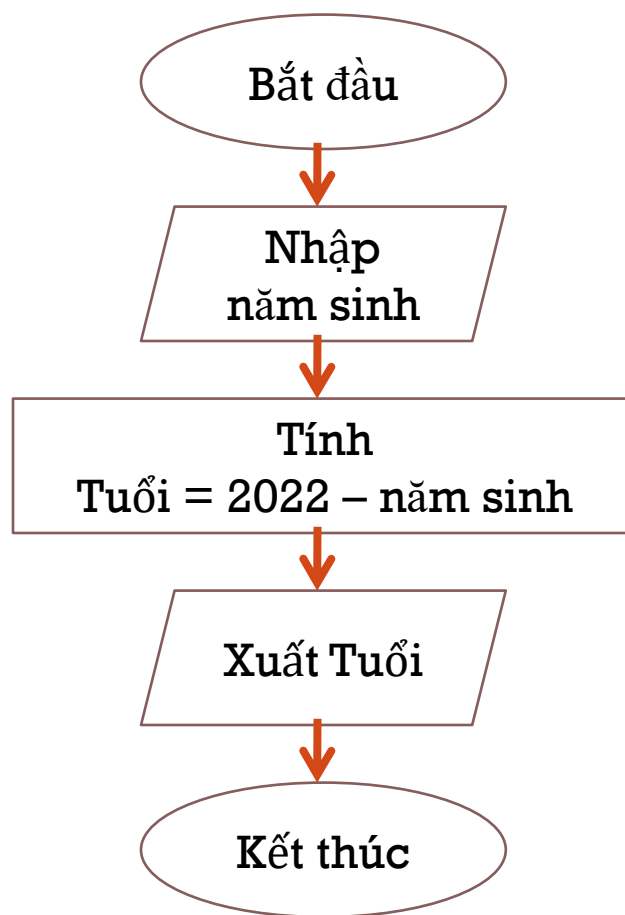


Bài tập thực hành

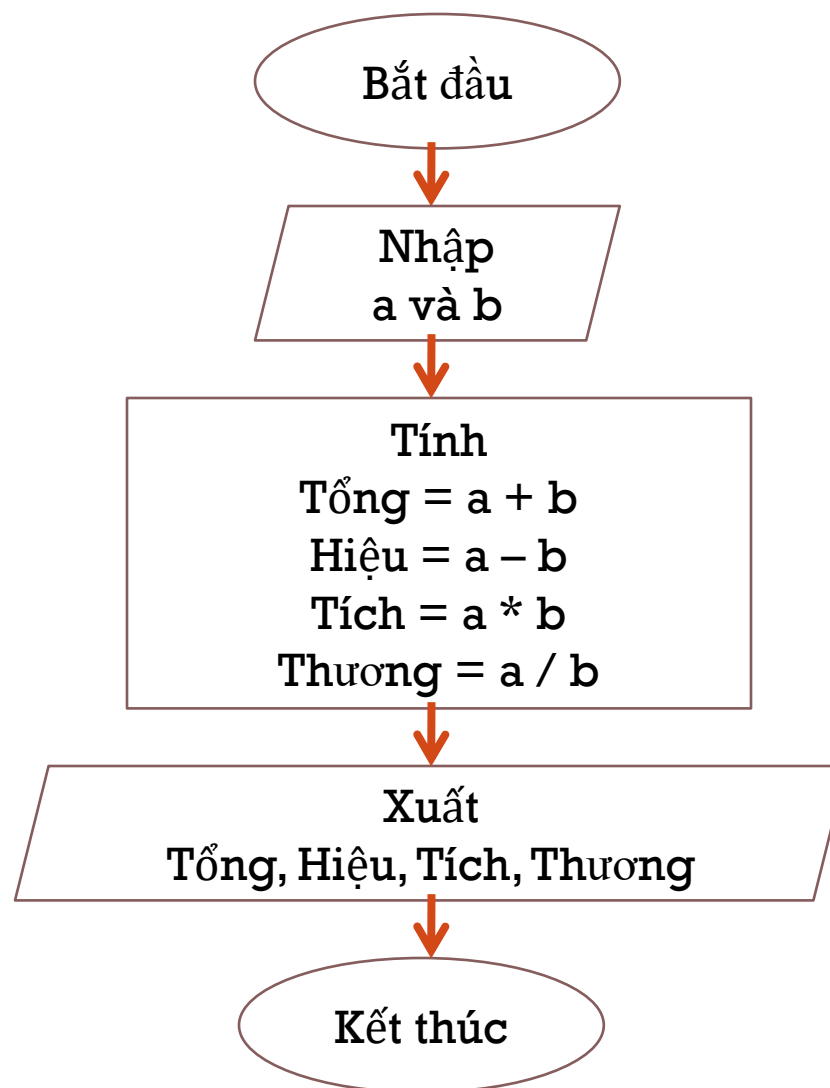
1. Nhập năm sinh của một người. Tính tuổi người đó.
2. Nhập 2 số a và b. Tính tổng, hiệu, tích và thương của hai số đó.
3. Nhập tên sản phẩm, số lượng và đơn giá. Tính tiền và thuế giá trị gia tăng phải trả, và tổng số tiền:
 1. $\text{tiền} = \text{số lượng} * \text{đơn giá}$
 2. $\text{thuế giá trị gia tăng} = 10\% \text{ tiền}$
4. Nhập bán kính của đường tròn. Tính chu vi và diện tích của hình tròn đó.
5. Nhập vào 2 số nguyên. Tính min và max của hai số đó.



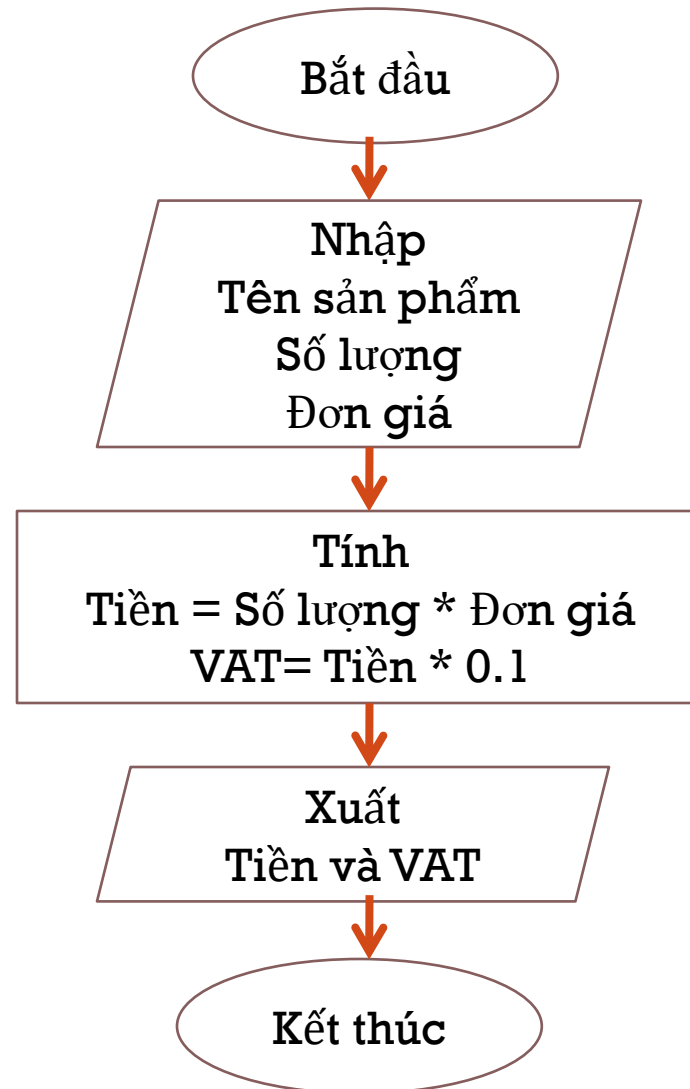
BÀI TẬP 1



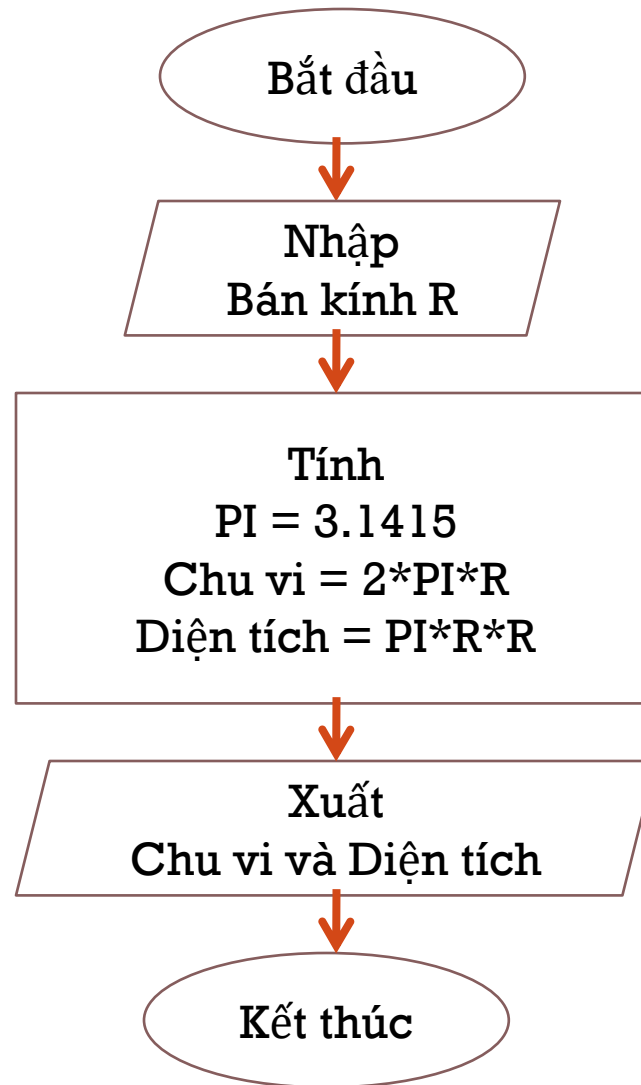
BÀI TẬP 2



BÀI TẬP 3



BÀI TẬP 4



BÀI TẬP 5

