**CHƯƠNG 1**

***Bài 1****.****Thảo luận nhóm***

-Phân biệt giữa ngôn ngữ máy, hợp ngữ và ngôn ngữ lập trình cấp cao.

**Ngôn ngữ máy :**  
- Mỗi loại máy tính đều có ngôn ngữ máy riêng của nó. Là ngôn ngữ duy nhất mà máy có thể trực tiếp hiểu và thực hiện.  
- Các lệnh viết bằng ngôn ngữ máy ở dạng mã nhị phân hoặc mã hexa  
\* Ưu điểm: khai thác triệt để tính năng phần cứng của máy tính, có thể trực tiếp hiểu được, không cần chương trình dịch.  
\* Nhược điểm: khó hiểu, khó nhớ, sử dụng nhiều câu lệnh để biểu diễn các thao tác.

**Hợp ngữ:**

Hợp ngữ là ngôn ngữ lập trình cấp thấp yêu cầu trình hợp dịch chuyển đổi thành mã máy mã đối tượng.

Con người có thể hiểu được ngôn ngữ hợp ngữ.

Hợp ngữ tuân theo một cú pháp tương tự như ngôn ngữ tiếng Anh.

Hợp ngữ bao gồm một tập hợp các lệnh chuẩn.

Hợp ngữ đang sử dụng cho các hệ thống thời gian thực dựa trên bộ vi xử lý.

**Ngôn ngữ bậc cao :**  
- Là ngôn ngữ gần với ngôn ngữ tự nhiên, có tính độc lập cao, ít phụ thuộc vào loại máy.  
\* Ưu điểm: dễ hiểu, dễ chỉnh sửa, tính độc lập cao.  
- Để máy tính hiểu được, cần phải có chương trình dịch để chuyển từ ngôn ngữ bậc cao sang ngôn ngữ máy.  
- Một số ngôn ngữ lập trình bậc cao: Pascal, C, Java, …

***Bài 2. Giới thiệu về một số ngôn ngữ lập trình thông dụng hiện nay về các nội dung:***

- Lịch sử phát triển;

- Đặc điểm;

- Ứng dụng

**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. ... Python được phát triển trong một dự án mã mở, do tổ chức phi lợi nhuận Python Software Foundation quản lý. Ban đầu, Python được phát triển để chạy trên nền Unix.

* Lịch sử phát triển:

Python 1: bao gồm các bản phát hành 1.x. Giai đoạn này, kéo dài từ đầu đến cuối thập niên 1990. Từ năm 1990 đến 1995, Guido làm việc tại CWI (*Centrum voor Wiskunde en Informatica* – Trung tâm Toán-Tin học) tại Amsterdam, Hà Lan. Vì nguyên nhân này, các phiên bản Python đầu tiên đều do CWI phát hành. Phiên bản cuối cùng phát hành tại CWI là 1.2.

Vào năm 1995, Guido chuyển sang Corporation for National Research Initiatives (CNRI) ở Reston, Virginia. Tại đây, ông phát hành một số phiên bản khác. Python 1.6 là phiên bản cuối cùng phát hành tại CNRI.

Sau bản phát hành 1.6, Guido rời bỏ CNRI để làm việc với các lập trình viên chuyên viết phần mềm thương mại. Tại đây, ông có ý tưởng sử dụng Python với các phần mềm tuân theo chuẩn GPL. Sau đó, CNRI và Free Software Foundation (FSF) đã cùng nhau hợp tác để làm bản quyền Python phù hợp với GPL. Cùng năm đó, Guido được nhận giải thưởng FSF Award for the Advancement of Free Software.

Phiên bản 1.6.1 ra đời sau đó là phiên bản đâu tiền tuân theo bản quyền GPL. Tuy nhiên, bản này hoàn toàn giống bản 1.6, trừ một số sửa lỗi cần thiết.

Python 2: vào năm 2000, Guido và nhóm phát triển Python dời đến BeOpen.com và thành lập BeOpen PythonLabs team. Phiên bản Python 2.0 được phát hành tại đây. Sau khi phát hành Python 2.0, Guido và các thành viên PythonLabs gia nhập Digital Creations.

* + Python 2.1 ra đời kế thừa từ Python 1.6.1 và Python 2.0. Bản quyền của phiên bản này được đổi thành Python Software Foundation License. Từ thời điểm này trở đi, Python thuộc sở hữu của Python Software Foundation (PSF), một tổ chức phi lợi nhuận được thành lập theo mẫu Apache Software Foundation.

Python 3: Về sự phát triển của Python trong tương lai, các nhà phát triển vẫn đang thảo luận về phiên bản mới: Python 3.0 (dự án gọi là Python 3000 hoặc Py3K). Dự kiến, dòng 3.x sẽ không hoàn toàn tương thích với dòng 2.x. Nhóm phát triển cho biết sẽ lấp những chỗ hở của ngôn ngữ. Nguyên tắc chủ đạo để phát triển Python 3.x là “bỏ cách làm việc cũ nhằm hạn chế trùng lặp về mặt chức năng của Python”. Tuy nhiên, cho đến nay vẫn chưa có kế hoạch cụ thể cho phát triển Python 3.

* Đặc điểm python:

### -Ngôn ngữ lập trình đơn giản, dễ học – dễ học

### **-Miễn phí, mã nguồn mở**

### **-Khả năng di động linh hoạt**

### **-Khả năng mở rộng và có thể nhúng**

### **-Ngôn ngữ thông dịch cấp cao**

### **-Thư viện tiêu chuẩn lớn để giải quyết những tác vụ phổ biến**

### **-Hướng đối tượng**

### **Uứng dụng python:**

### -Nghiên cứu mang tính học thuật với dữ liệu lớn

### -Lập trình Các ứng dụng web

### -Ngành khoa học dữ liệu (data science)

### -Machine Learning và Trí thông minh nhân tạo (AI)

### -Lĩnh vực IoT – Internet Vạn Vật

### -Lập trình game.

**CHƯƠNG 1**

### GIỚI THIỆU VỀ MÁY TÍNH VÀ LẬP TRÌNH

### THẢO LUẬN NHÓM

### *Bài 1. Xác định vai trò của thuật toán trong lập trình.*

* Thuật toán là một quy trình bao gồm các quy tắc cụ thể để giải quyết các vấn đề được đề ra trong phạm bị hữu hạn. Mục đích của thuật toán là tìm ra kết quả giải quyết dữ kiện, vấn đề ban đầu.
* Vai trò:

-Thuật toán là phần quan trọng, không thể thiếu khi tiếp cận các vấn đề liên quan đến lĩnh vực lập trình.

-Thuật toán tốt mang đến hiệu quả cao, giúp các chương trình hoạt động hiệu quả với tốc độ xử lý nhanh chóng, tiết kiệm tài nguyên.

-Thuật toán giúp lập trình viên hiểu rõ và sâu hơn về ứng dụng, chương trình.

***Bài 2. Khái niệm về mã giả và cách sử dụng mã giả***

* Khái niệm:

-Là một bản mô tả [giải thuật](https://vi.wikipedia.org/wiki/Thu%E1%BA%ADt_to%C3%A1n" \o "Thuật toán) [lập trình máy tính](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_m%C3%A1y_t%C3%ADnh" \o "Lập trình máy tính) ngắn gọn và không chính thức cấp cao, trong đó sử dụng những quy ước có cấu trúc của một số [ngôn ngữ lập trình](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh" \o "), nhưng thường bỏ đi những chi tiết không cần thiết để giúp hiểu rõ giải thuật hơn, như bỏ đi [chương trình con](https://vi.wikipedia.org/wiki/Ch%C6%B0%C6%A1ng_tr%C3%ACnh_con" \o "Ngôn ngữ tự nhiên), [khai báo biến](https://vi.wikipedia.org/wiki/Khai_b%C3%A1o_bi%E1%BA%BFn" \o ") và những đoạn mã đặc biệt của hệ thống

* Cách sử dụng:

-Luôn viết hoa từ đầu tiên (thường là một trong 6 cấu trúc chính).

-Chỉ có một câu lệnh trên mỗi dòng.

**-**Thụt lề để hiển thị thứ bậc, cải thiện khả năng đọc và hiển thị các cấu trúc lồng nhau.

-Luôn kết thúc các phần nhiều dòng bằng cách sử dụng bất kỳ từ khóa END nào ( *ENDIF* , *ENDWHILE* , v.v.).

-Giữ cho ngôn ngữ lập trình câu lệnh của bạn độc lập .

-Sử dụng miền đặt tên của vấn đề , không phải miền triển khai . Ví dụ: “ *Nối họ với tên* ” thay vì “ *name = first + last.*”

-Giữ cho nó đơn giản , ngắn gọn và dễ đọc .

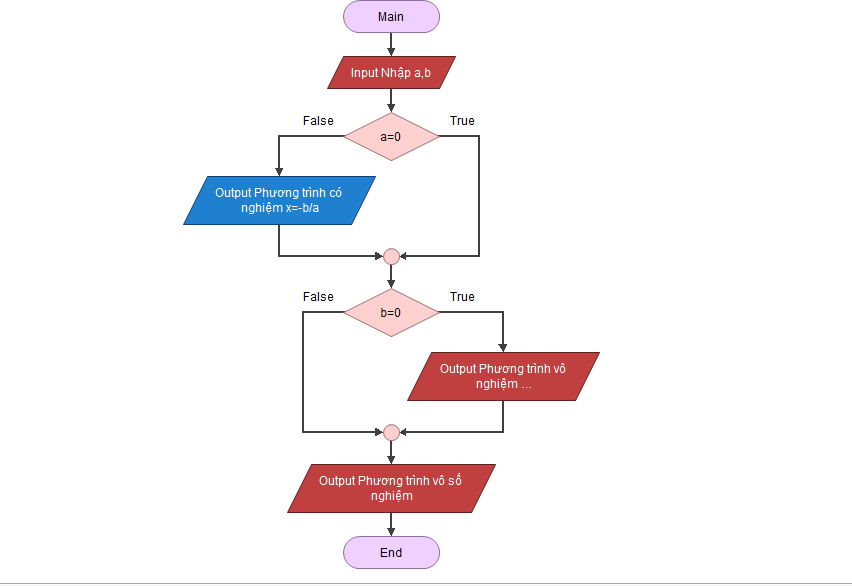
**BÀI TẬP**

. Sử dụng phần mềm Flowgorithm để thiết kế các thuật toán:

- Giải phương trình bậc nhất;

- Giải phương trình bậc 2;

-Giải phương trình bậc nhất:



Giải phương trình bậc 2;

