



به نام خالق خرد

# مبانی برنامه سازی FUNDAMENTALS OF PROGRAMMING





مدرس: بی نظیر بیگلری

جلسه ۱: مفاهیم اولیه

- ① آشنایی با کلیت درس
  - ① سیلابس
  - ① نحوه ارزیابی
  - ① نحوه اطلاع رسانی
- ① تعاریف اولیه
- ① معرفی اجزای اصلی کامپیوتر
- ① تاریخچه رشد زبان های برنامه سازی
- ① آشنایی با زبان پایتون
- ① مراحل ساخت و اجرای یک برنامه

# آشنایی با کلیت درس


## هدف:

- آشنایی با مبانی برنامه‌سازی کامپیوتر 
- ایجاد تفکر الگوریتمی در حل مسائل 
- کسب توانایی پیاده‌سازی الگوریتم‌ها به وسیله کامپیوتر 
- آشنایی با اصول اولیه نوشتن برنامه‌های ساخت‌یافته و مهندسی‌ساز 

## مراجع:

اصلی: 

How to Think Like a Computer Scientist: Learning with Python. 3<sup>rd</sup> Ed. 2011

کمکی: 

Practical Programming: An Introduction to Computer Science Using Python. 2009

Python Programming: An Introduction to Computer Science. 2004

- ④ تمرین برنامه‌سازی + یک پروژه دو یا سه فازی ۶ نمره
- ④ آزمونک‌ها ۲ نمره
- ④ آزمون‌ها (میان‌ترم و پایان‌ترم) ۱۲ نمره
- ④ مسابقه برنامه‌نویسی به صورت آن‌لاین ۱+ نمره

**امتحان میان‌ترم      هنوز مشخص نشده‌است**

**امتحان پایان‌ترم      دوشنبه ۲۶ خرداد**

(عدم حضور در کلاس تأثیر منفی خواهد داشت)

# نحوه اطلاع رسانی

- ① مرجع اصلی، صفحه درس ۴۰۱۵۳ سایت دانشکده مهندسی کامپیوتر (بخش [ce.sharif.ir/courses](http://ce.sharif.ir/courses) گروه ۷).
- ② ثبت نام در سایت Piazza
- ③ دانشکده مهندسی کامپیوتر- اتاق ۶۱۲ آزمایشگاه مهندسی متدولوژی (روزهای ۲شنبه و ۴شنبه) البته از طریق ایمیل یا Piazza از قبل باید هماهنگ شود.
- ④ سردستیار آموزشی: خانم نگار ابوالحسنی

- ◎ الگوریتم: روش انجام یک کار با ذکر دقیق تمام مراحل آن به طوریکه ترتیب انجام کارها، شروع و پایان آن مشخص باشد؛ به عبارت دقیق‌تر هر دستورالعملی که مراحل مختلف کاری را به زبان دقیق و با جزئیات کافی بیان نماید و در آن ترتیب مراحل و خاتمه‌پذیر بودن عملیات کاملاً مشخص باشد الگوریتم نامیده می‌شود.
- ◎ فلوچارت: نمایش الگوریتم به صورت شماتیک
- ◎ حل مسئله: شامل شناخت مسئله، طرح نقشه حل مسئله و تحلیل راه حل مسئله (اجرا، واریسی یا تعمیم راه حل)
- ◎ برنامه: نمود الگوریتم به کمک دستورات زبان برنامه‌سازی

- ◎ مهارت قدرت شناخت دقیق مسئله (تعیین داده‌ها، مجهول‌ها و رابطه بین آنها)
- ◎ مهارت ایجاد طرحی برای حل مسئله به صورت الگوریتمیک (تفکر خلاقانه در مورد راه حل و بیان واضح و دقیق راه حل)
- ◎ اجرای راه حل و اطمینان از درستی راه حل

# اجزای اصلی الگوریتم

شروع

دستورات ورودی/ خروجی: گرفتن مقدار یک عدد، کاراکتر، رشته و ... از کاربر/ نمایش خروجی روی مانیتور یا چاپ روی پرینتر و ...

دستورات محاسباتی، مقداردهی: انجام محاسبه مشخص، ذخیره کردن یک مقدار در یک متغیر

دستورات تصمیم‌گیری: ارزیابی یک شرط ساده یا مرکب

پرش

برو به «گام؟» .

پایان

«مقدار؟» را از کاربر بگیر.

«پیام؟» را به کاربر نشان بده.

«مقدار؟» را در «متغیر؟» بریز.

اگر «شرط؟» آنگاه «دستور؟» وگرنه «دستور؟».



# الگوریتم حاصل جمع دو عدد

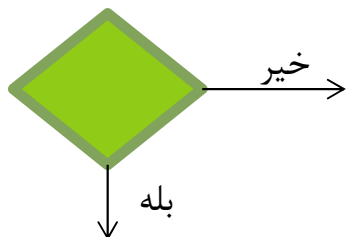
1. شروع کن.
2. عدد  $a$  و عدد  $b$  را از کاربر بگیر.
3.  $a$  را با  $b$  جمع کن و حاصل آن را در  $c$  بریز.
4.  $c$  را چاپ کن.
5. پایان.



دستورات ورودی / خروجی



مقداردهی و محاسبات



تصمیم‌گیری

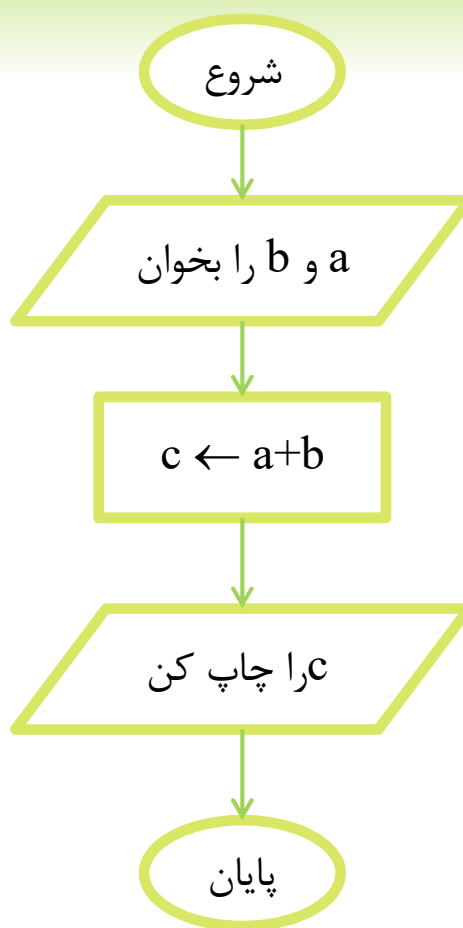


پررش



شروع و پایان

# فلوچارت حاصل جمع دو عدد



# معرفی اجزای اصلی کامپیوتر

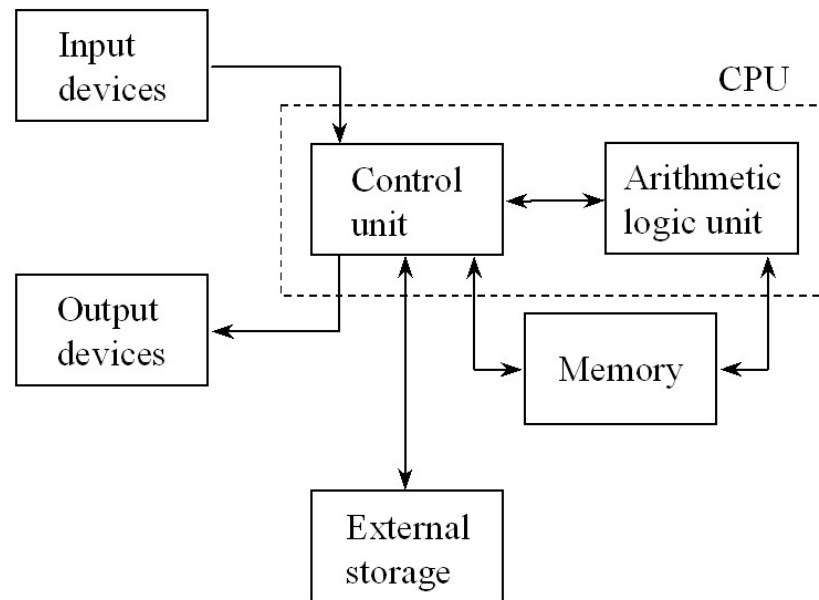
## دیدگاه اول:

1. سخت افزار: دستگاه هایی که کامپیوتر را می سازند مثل واحدهای پردازنده، حافظه، صفحه کلید و ...
2. نرم افزار: برنامه هایی که روی کامپیوتر اجرا می شوند.

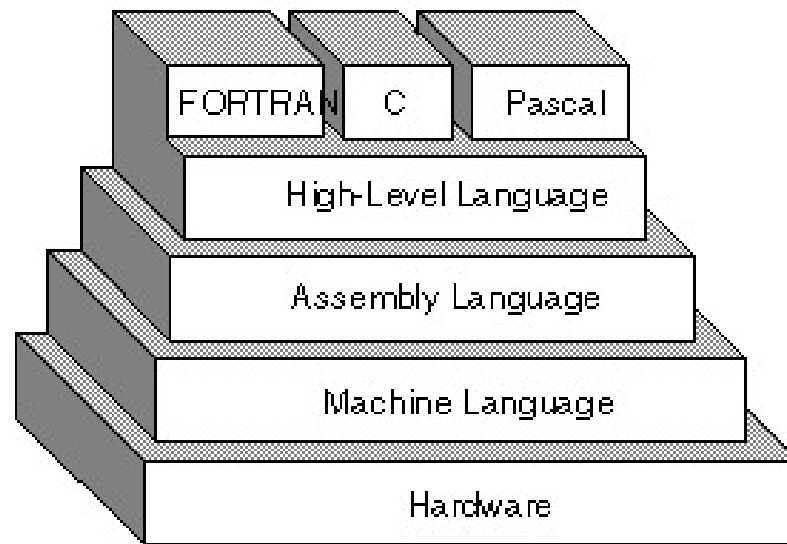
## دیدگاه دوم:

1. واحدهای ورودی و خروجی
2. واحد حافظه
3. واحد محاسبه و منطق (Arithmetic and Logic Unit)
4. واحد پردازش مرکزی (Central Processing Unit)
5. واحد ذخیره جانبی

# سازماندهی اجزاء کامپیوتر



# جایگاه زبان های برنامه سازی



- ◎ زبان مصنوعی برای دستور دادن به دستگاه‌های برنامه‌پذیر مثل کامپیوترهاست.
- ◎ سیستم‌های عامل، منابع مختلف را مثل CPU، حافظه و... را به برنامه‌های مختلف اختصاص می‌دهند؛ بنابراین واسطه بین برنامه‌ها (نرم‌افزارها) و سخت‌افزارها هستند یا در حقیقت مدیریت تقسیم منابع (سخت‌افزار) بین نرم‌افزارها بر عهده سیستم عامل است.
- ◎ سیستم‌های عامل، نرم‌افزارها و حتی سخت‌افزارها با استفاده از زبان‌های برنامه‌سازی نوشته و تولید می‌شوند.
- ◎ زبان ماشین و یا زبان اسمبلی سطح پایین هستند چون به سخت‌افزار نزدیک‌اند.
- ◎ زبانی مثل Python یا C سطح بالا هستند و باید قبل از اجرا به چیزی مثل زبان ماشین ترجمه شوند.
- ◎ علت وجود زبان‌های سطح بالا: سادگی، کاهش اندازه برنامه، خوانایی بیشتر، احتمالاً درست‌تر و قابلیت حمل

# آشنایی بیشتر با زبان PYTHON

- 🎯 زبان پایتون بسیار ساده است و برای رشته‌های غیر از کامپیوتر مناسب.
- 🎯 یک موتور که برنامه‌های پایتون را ترجمه می‌کند Python Interpreter
- 🎯 دو راه استفاده:

1. Immediate Mode

2. Script Mode

- 🎯 به `>>>` Python Prompt گفته می‌شود که نشان می‌دهد مترجم برای دریافت دستورات آماده است.



# نوشتن اولین برنامه

- ◎ کوتاه‌ترین برنامه به طور سنتی در برنامه‌سازی، برنامه‌ای است که یک عبارت را در صفحه نمایش چاپ کند (Hello World!)
- ◎ در پایتون کافی است در یک فایل با پسوند `.py` خط زیر را بنویسید:  
  
◎ `print("Hello World!")`
- ◎ برای اجرای آن در حال حاضر از ابزار IDLE استفاده می‌کنیم (از صفحه درس قسمت Links می‌توانید دانلود کنید Python 3.3 یا نسخه Python 3.2)

## مقدمات برنامه‌سازی