مبانی کامپیوتر و برنامه سازی

دانشگاه آل طه

رشته کامپیوتر

درس تخصصی

پاییز ۱۴۰۳

رئوس مطالب

- مفاهیم اولیه
- محاسبات در کامپیوتر
- آشنایی با الگوریتم و فلوچارت
- مقدمات برنامه سازی
- فرمت بندی ورودی/خروجی
- دستورها
- توابع
- آزمون و خطایابی برنامه

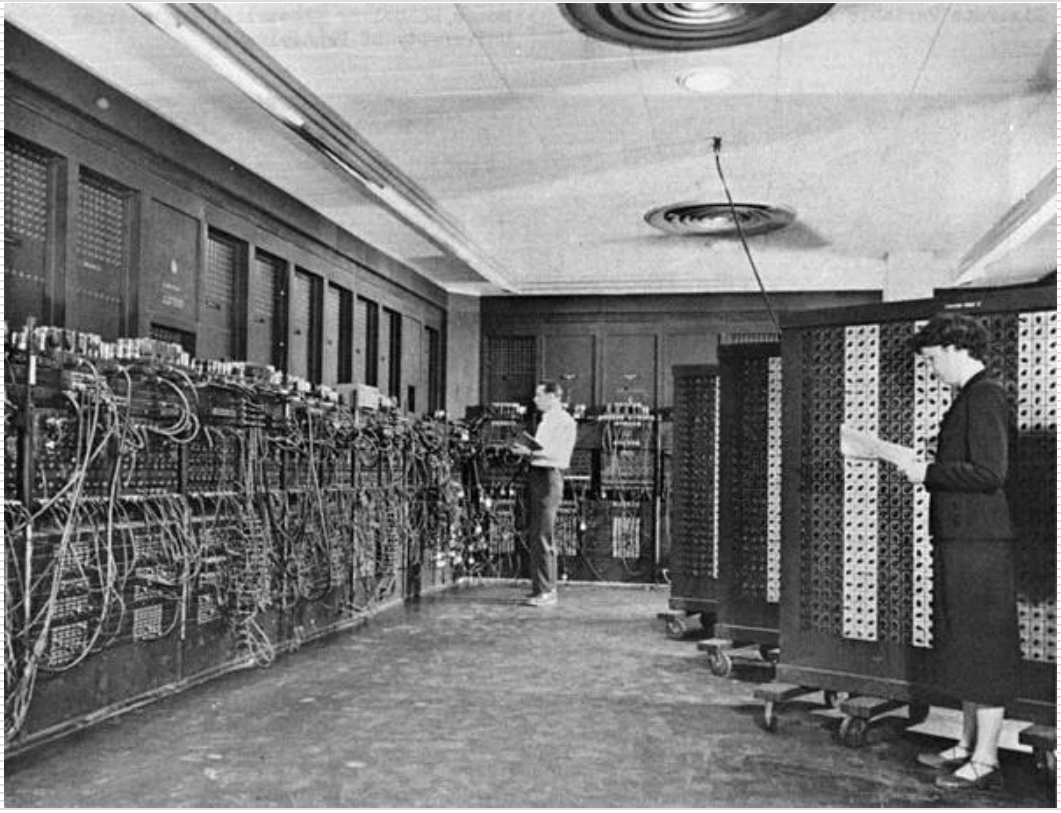
- آرایه ها
- اشاره گرها
- کاراکترها و رشته ها
- ساختارها
- آشنایی با c++
- آشنایی با رده ها
- حوزه های private , public
- پرونده ها

جلسه اول

- تاریخچه توسعه کامپیوتر : سیستم های عامل ، زبان های برنامه سازی
- معرفی اجزای اصلی کامپیوتر
- تاریخچه c++
- برنامه سازی ساخت یافته و نوشتن کد مهندسی ساز
- ساختار و مراحل ساخت و اجرای برنامه ، آشنایی کلان با کامپایلر

تاریخچه توسعه کامپیوتر : سیستم های عامل ، زبان های برنامه سازی

- تعریف کامپیوتر :ماشینی برای انجام محاسبات



تاریخچه

دوره نسل اول: ۱۹۴۶-۱۹۵۹. بر پایه لوله خلاء.

- فناوری لامپ الکترونی (Vacuum tube technology)
- نامطمئن (Unreliable)
- فقط زبان ماشین را پشتیبانی می‌کرد (Supported machine language only)
- بسیار پرهزینه (Very costly)
- حرارت زیادی تولید می‌کرد (Generated a lot of heat)
- دستگاه‌های ورودی و خروجی‌اش کند بود (Slow input and output devices)
- اندازه بسیار بزرگ (Huge size)
- نیاز به جریان متناوب برق (Need of AC)
- غیر قابل حمل (Non-portable)
- مصرف بسیار بالای برق (Consumed a lot of electricity)

تاریخچه توسعه کامپیوتر : سیستم های عامل ، زبان های برنامه سازی

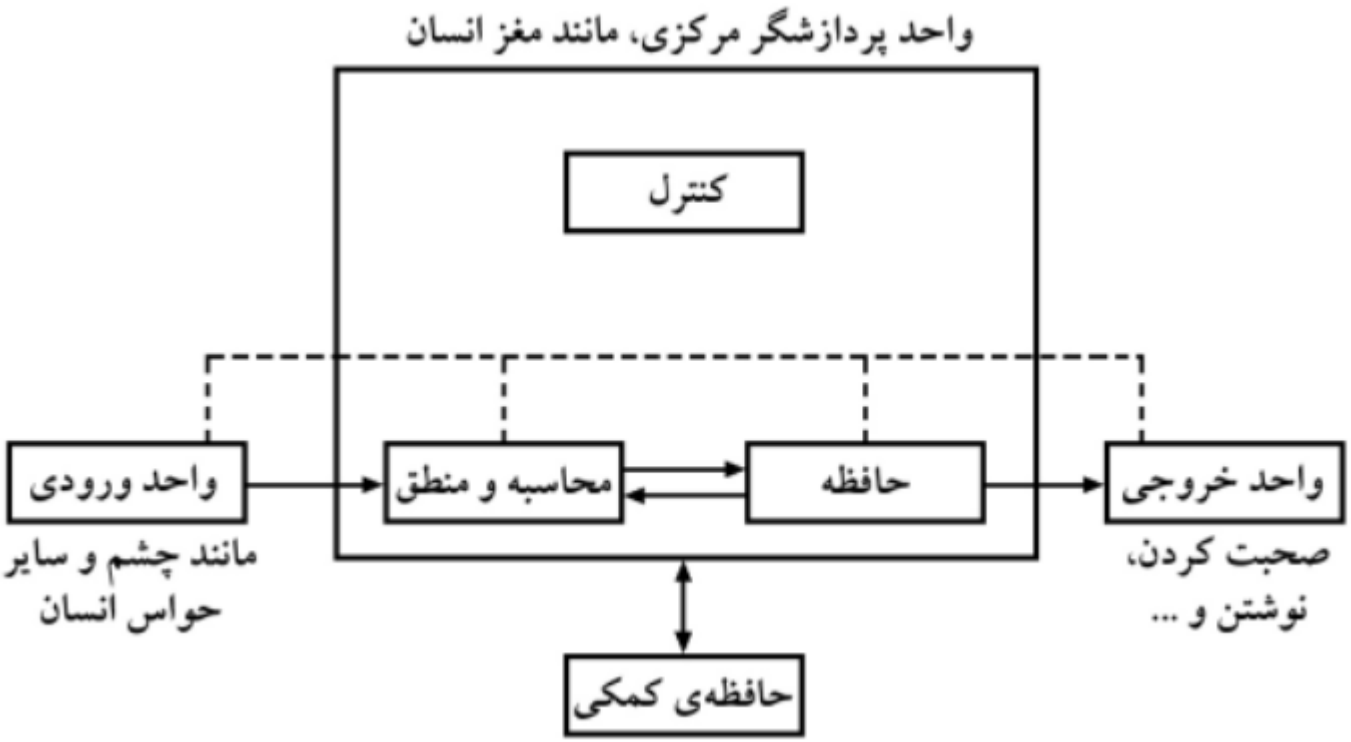
- دوره نسل اول: ۱۹۴۶-۱۹۵۹. بر پایه لوله خلاء.
- دوره نسل دوم: ۱۹۵۹-۱۹۶۵. بر پایه ترانزیستور.
- دوره نسل سوم: ۱۹۶۵-۱۹۷۱. بر پایه مدار یکپارچه.
- دوره نسل چهارم: ۱۹۷۱-۱۹۸۰. بر پایه میکروپردازنده .VLSI
- دوره نسل پنجم: ۱۹۸۰ به بعد. بر پایه میکروپردازنده .ULSI

تاریخچه توسعه کامپیوتر : سیستم های عامل ، زبان های برنامه سازی

توضیح اجزای اصلی یک کامپیوتر

- **واحد ورودی**(Input Unit): **اطلاعات را از دنیای خارج دریافت می‌کند.**
- **واحد خروجی**(Output Unit): **نتایج را به کاربر نمایش می‌دهد.**
- **واحد حافظه**(Memory Unit): **اطلاعات را ذخیره می‌کند.**
- **واحد محاسبه و منطق**(Arithmetic Logic Unit - ALU): **محاسبات و عملیات منطقی را انجام می‌دهد.**
- **واحد کنترل**(Control Unit): **سایر واحدها را کنترل می‌کند.**

تاریخچه توسعه کامپیوتر : سیستم های عامل ، زبان های برنامه سازی



حافظه RAM و ROM

- RAM (Random Access Memory)یا حافظه دسترسی تصادفی
  - سرعت بالا: RAMسریع‌ترین نوع حافظه در کامپیوتر است و اطلاعات به سرعت در آن خوانده و نوشته می‌شوند.
  - حافظه موقت: اطلاعات ذخیره شده در RAMبا قطع برق پاک می‌شوند.
  - استفاده: برای اجرای برنامه‌ها و ذخیره داده‌های در حال استفاده به صورت موقت استفاده می‌شود.

حافظه RAM و ROM

- ROM (Read Only Memory)یا حافظه فقط خواندنی
  - حافظه دائمی: اطلاعات ذخیره شده در ROMحتی پس از قطع برق نیز حفظ می‌شود.
  - فقط خواندنی: اطلاعات در ROMفقط قابل خواندن هستند و نمی‌توان آن‌ها را به راحتی تغییر داد.
  - استفاده: برای ذخیره اطلاعاتی که نیازی به تغییر ندارند، مانند BIOS)برنامه اولیه ورودی/خروجی) که کامپیوتر را هنگام روشن شدن راه اندازی می‌کند، استفاده می‌شود.

مقایسه حافظه RAM و ROM

ویژگی	RAM	ROM
سرعت	بسیار بالا	نسبتاً پایین
نوع	فرار (Volatile)	غیر فرار (Non-Volatile)
عملکرد	ذخیره موقت داده‌ها برای اجرای برنامه‌ها	ذخیره دائمی اطلاعات ضروری برای راه اندازی سیستم
قابلیت تغییر	قابل تغییر	معمولاً قابل تغییر نیست
مثال	حافظه رم در کامپیوتر	BIOS، حافظه فلش در برخی دستگاه‌ها

واحد های اندازه گیری حافظه

واحد	برابر با	کاربرد متداول
بیت (Bit)	کوچکترین واحد اطلاعات	
بایت (Byte)	۸ بیت	اندازه یک کاراکتر
کیلوبایت (KB)	۱۰۲۴ بایت	فایل‌های کوچک (سند، عکس کوچک)
مگابایت (MB)	۱۰۲۴ کیلوبایت	فایل‌های متوسط (موسیقی، عکس با کیفیت بالا)
گیگابایت (GB)	۱۰۲۴ مگابایت	هارد دیسک، حافظه فلش
ترابایت (TB)	۱۰۲۴ گیگابایت	هارد دیسک‌های بزرگ، مراکز داده
پتابایت (PB)	۱۰۲۴ ترابایت	مراکز داده بسیار بزرگ
اگزابایت (EB)	۱۰۲۴ پتابایت	کل داده‌های موجود در اینترنت

مثال

کامپیوتری دارای ۳۲ گیگابایت حافظه‌ی جانبی است. این کامپیوتر دارای چند مگابایت، کیلوبایت، بایت و بیت حافظه است؟

$1\ GB = 1024\ MB$, $32\ GB = 32 \times 1024 = 2^5 \times 2^{10} = 2^{15}\ MB$

$1\ GB = 1024 \times 1024\ KB$, $32\ GB = 32 \times 1024 \times 1024 = 2^5 \times 2^{10} \times 2^{10} = 2^{25}\ KB$

$1\ GB = 1024 \times 1024 \times 1024\ B$, $32\ GB = 32 \times 1024 \times 1024 \times 1024 = 2^5 \times 2^{10} \times 2^{10} \times 2^{10} = 2^{35}\ B$

$1\ GB = 1024 \times 1024 \times 1024 \times 8\ Bit$, $32\ GB = 32 \times 1024 \times 1024 \times 1024 \times 8 = 2^5 \times 2^{30} \times 2^3 = 2^{38}\ Bit$

Operating System سیستم عامل

- نرم‌افزاری است که مدیریت منابع کامپیوتر را به عهده دارد و بستری را فراهم می‌سازد که نرم‌افزار کاربردی اجرا شده و از خدمات آن استفاده کنند.
  - مدیریت منابع: تخصیص بهینه حافظه، پردازنده و دیسک سخت به برنامه‌ها.
  - اجرای برنامه‌ها: بارگذاری و اجرای برنامه‌ها توسط پردازنده.
  - تعامل با کاربر: فراهم کردن رابط کاربری برای برقراری ارتباط با کاربر.
  - مدیریت فایل‌ها: سازماندهی و مدیریت فایل‌ها در دیسک سخت.
  - ارائه خدمات: ارائه خدمات جانبی به برنامه‌ها برای انجام وظایف پیچیده.

سیستم عامل

- در روزهای اولیه کامپیوترها، همه چیز به صورت دستی انجام می‌شد؛ برنامه‌نویسان مجبور بودند دستورها را به صورت مستقیم وارد کنند و سخت‌افزار را مدیریت کنند. این فرآیند بسیار پیچیده و زمان‌بر بود.
- مزایای استفاده از سیستم عامل‌ها:
  - سادگی استفاده: سیستم عامل‌ها با ارائه یک رابط کاربری ساده، امکان تعامل آسان با کامپیوتر را برای کاربران فراهم کردند.
  - افزایش بهره‌وری: سیستم عامل‌ها با مدیریت منابع کامپیوتر به صورت خودکار، بهره‌وری را افزایش دادند و امکان اجرای همزمان چندین برنامه را فراهم کردند.
  - امنیت: سیستم عامل‌ها با ارائه مکانیزم‌های امنیتی، از داده‌های کاربران محافظت می‌کنند.
  - سازگاری: سیستم عامل‌ها با ارائه یک محیط استاندارد، امکان اجرای انواع مختلف نرم‌افزارها را فراهم می‌کنند.

سیستم عامل

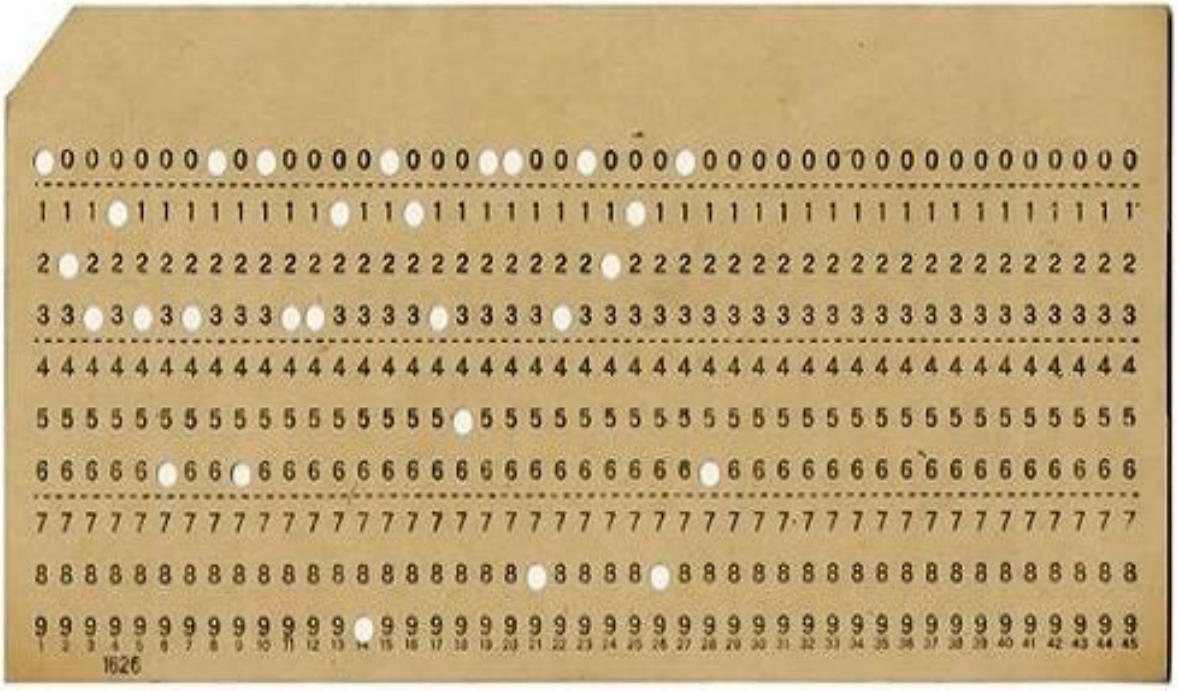
- **سیستم‌های دسته‌ای (Batch Systems) دهه ۱۹۵۰**
  - **عدم تعامل مستقیم: کاربران نمی‌توانستند در حین اجرای برنامه با کامپیوتر تعامل داشته باشند.**
  - **اجرای دسته ای: برنامه‌ها به صورت یک دسته و به ترتیب مشخصی اجرا می‌شدند.**
  - **اپراتور انسانی: یک اپراتور انسانی وظیفه جمع‌آوری برنامه‌ها، آماده‌سازی آن‌ها برای اجرا و ارسال آن‌ها به کامپیوتر را بر عهده داشت.**
  - **کارت‌های پانچ: معمولاً از کارت‌های پانچ برای ورود داده‌ها و برنامه‌ها به کامپیوتر استفاده می‌شد.**
  - **زمان انتظار طولانی: از آنجا که برنامه‌ها به نوبت اجرا می‌شدند، کاربران برای دریافت نتایج برنامه‌های خود باید مدت زیادی صبر می‌کردند.**

سیستم عامل

- **سیستم‌های دسته‌ای (Batch Systems) دهه ۱۹۵۰**
  - **عدم تعامل مستقیم: کاربران نمی‌توانستند در حین اجرای برنامه با کامپیوتر تعامل داشته باشند.**
  - **اجرای دسته ای: برنامه‌ها به صورت یک دسته و به ترتیب مشخصی اجرا می‌شدند.**
  - **اپراتور انسانی: یک اپراتور انسانی وظیفه جمع‌آوری برنامه‌ها، آماده‌سازی آن‌ها برای اجرا و ارسال آن‌ها به کامپیوتر را بر عهده داشت.**
  - **کارت‌های پانچ: معمولاً از کارت‌های پانچ برای ورود داده‌ها و برنامه‌ها به کامپیوتر استفاده می‌شد.**
  - **زمان انتظار طولانی: از آنجا که برنامه‌ها به نوبت اجرا می‌شدند، کاربران برای دریافت نتایج برنامه‌های خود باید مدت زیادی صبر می‌کردند.**

سیستم عامل

- تصویر کارت پانچ
  - موقعیت سوراخ‌ها: هر سوراخ در یک موقعیت خاص، بیانگر یک بیت اطلاعات است.



سیستم عامل

- دهه ۱۹۶۰: سیستم‌های اشتراک زمانی (Time-Sharing Systems)
  - چندکاربر بودن: استفاده همزمان چند کاربر
  - تعامل مستقیم: تعامل مستقیم کاربر با کامپیوتر بدون نیاز به اپراتور
  - تقسیم زمان پردازنده: تقسیم زمان پردازنده به صورت قطعه‌های کوچک بین برنامه‌های مختلف
  - مزایا : سرعت بیشتر ، تعامل بیشتر ، استفاده بهینه از منابع

سیستم عامل

- **IBM 7094: یکی از غول‌های محاسباتی دهه ۶۰**
  - **ناسا از آن برای کنترل پروازهای فضایی مرکوری و جمنای استفاده کرد**

سیستم عامل

- ۴. دهه ۱۹۷۰ تا ۱۹۹۰: توسعه سیستم‌های چندکاربره و شبکه‌ای (مانند UNIXو Windows)
  - چندکاربری: امکان اجرای همزمان چندین برنامه توسط چندین کاربر به صورت مستقل.
  - چندوظیفه‌ای: امکان اجرای همزمان چندین برنامه توسط یک کاربر.
  - مدیریت منابع: تخصیص منابع سیستم (مانند حافظه، پردازنده و دستگاه‌های ورودی/خروجی) به صورت بهینه بین کاربران و برنامه‌ها.
  - شبکه‌ای: امکان اتصال به شبکه‌های کامپیوتری و اشتراک‌گذاری داده‌ها و منابع با سایر کامپیوترها.
  - امنیت: محافظت از داده‌ها و سیستم در برابر دسترسی‌های غیرمجاز.

سیستم عامل

- ۴. دهه ۱۹۷۰ تا ۱۹۹۰: توسعه سیستم‌های چندکاربره و شبکه‌ای (مانند UNIXو Windows)
- تأثیر این تحولات:
  - انقلاب در محاسبات: این تحولات منجر به انقلاب در نحوه استفاده از کامپیوترها شد. کامپیوترها از ابزارهای تخصصی به ابزارهای همه منظوره تبدیل شدند.
  - رشد اینترنت: سیستم‌های چندکاربره و شبکه‌ای نقش مهمی در توسعه اینترنت و وب جهانی ایفا کردند.
  - توسعه نرم‌افزارهای کاربردی: با وجود سیستم‌های عامل قدرتمند، توسعه نرم‌افزارهای کاربردی متنوعی مانند پردازشگرهای متن، صفحات گسترده، پایگاه داده‌ها و نرم‌افزارهای گرافیکی امکان‌پذیر شد.

زبان‌های برنامه‌سازی

زبان برنامه سازی ، پل ارتباطی بین انسان و کامپیوتر

- زبان‌های ابتدایی: اسمبلی و ماشین.
- نسل اول: زبان ماشین
  - زبان مستقیم کامپیوتر: رشته‌ای از اعداد صفر و یک که برای کامپیوتر قابل فهم بود اما برای انسان بسیار پیچیده و دشوار بود.
  - مشکل در یادگیری و استفاده: بسیار وقت‌گیر و مستعد خطا

زبان‌های برنامه‌سازی

زبان برنامه سازی ، پل ارتباطی بین انسان و کامپیوتر

- زبان‌های ابتدایی: اسمبلی و ماشین.
- نسل دوم: زبان اسمبلی
  - سطح بالاتر از زبان ماشین: زبان اسمبلی یک قدم به سمت انسان نزدیک‌تر شد. به جای اعداد از کلمات کوتاه و قابل فهم‌تری (مانند ADDبرای جمع) استفاده می‌شد.
  - هنوز وابسته به سخت‌افزار: به معماری سخت‌افزاری کامپیوتر وابسته بود و برای هر نوع پردازنده، زبان اسمبلی متفاوتی وجود داشت.

زبان‌های برنامه‌سازی

ظهور زبان‌های سطح بالا

ساختارهای پیچیده‌تر و نزدیک‌تر به زبان انسان

- FORTRAN (1957): اولین زبان برنامه‌سازی سطح بالا برای محاسبات علمی.
- COBOL (1959): برای برنامه‌های تجاری.
- C (1972): مهم‌ترین زبان سیستم‌های عامل.
- C++ (1983): معرفی شی‌گرایی به C.

زبان‌های برنامه‌سازی

ویژگی‌های کلیدی C++

- شیءگرایی: C++به برنامه‌نویسان اجازه می‌دهد تا برنامه‌های خود را به صورت مجموعه‌هایی از اشیاء مدل‌سازی کنند که هر کدام دارای ویژگی‌ها و رفتارهای خاص خود هستند. این رویکرد باعث می‌شود کدها قابل فهم‌تر، قابل نگهداری‌تر و قابل توسعه‌تر شوند.
- سازگاری با C: C++به عنوان یک توسعه‌ی زبان Cطراحی شده است، بنابراین برنامه‌نویسان Cمی‌توانند با تغییرات اندکی کدهای خود را به C++منتقل کنند.

زبان‌های برنامه‌سازی

ویژگی‌های کلیدی C++

- کنترل حافظه: C++به برنامه‌نویسان امکان می‌دهد تا به صورت مستقیم بر مدیریت حافظه سیستم نظارت داشته باشند. این ویژگی به آن‌ها اجازه می‌دهد تا برنامه‌های بسیار کارآمد و بهینه بنویسند، اما در عین حال خطایابی را نیز پیچیده‌تر می‌کند.
- کارایی بالا: به دلیل ساختار نزدیک به زبان ماشین، برنامه‌های نوشته شده به زبان C++ معمولاً سرعت اجرای بالایی دارند و برای توسعه نرم‌افزارهای سیستم و بازی‌ها بسیار مناسب هستند.

برنامه‌سازی ساخت‌یافته

تعریف: برنامه‌سازی ساخت‌یافته ((Structured Programmingیک روش برنامه‌نویسی است که بر پایه اصول تقسیم برنامه به بخش‌های کوچک‌تر و قابل مدیریت استوار است.

این روش کد را به توابع و ماژول‌ها تقسیم می‌کند که هر کدام وظیفه خاصی را انجام می‌دهند. این بخش‌ها به صورت مجزا توسعه و تست می‌شوند و سپس در کنار هم قرار می‌گیرند.

- **نکات کلیدی:**
  - **هر تابع یا ماژول یک وظیفه مستقل را برعهده دارد.**
  - **استفاده از توابع کنترلی مانند حلقه‌ها و شرط‌ها به جای دستورهای .GOTO**
  - **اجرای برنامه به صورت خطی و قابل پیش‌بینی.**

مفهوم توابع و ماژولاریتی در برنامه‌سازی

- **توابع:**
  - **توابع بخش‌های کوچکی از کد هستند که وظایف خاصی را انجام می‌دهند و می‌توانند در نقاط مختلف برنامه فراخوانی شوند.**
  - **هر تابع ورودی می‌گیرد، پردازش می‌کند و نتیجه‌ای را برمی‌گرداند (اگر نیاز باشد).**

- **ماژولاریتی (Modularity) :**
  - **ماژولاریتی به معنای تقسیم برنامه به بخش‌های مستقل (ماژول‌ها) است که هر کدام قابلیت اجرا و توسعه جداگانه دارند.**
  - **هر ماژول می‌تواند شامل چندین تابع یا زیرسیستم باشد.**

توابع و ماژولاریتی در برنامه‌سازی

مزایا:

- **استفاده مجدد از کد: توابع و ماژول‌ها قابل استفاده مجدد در بخش‌های دیگر برنامه یا پروژه‌های دیگر هستند.**
- **کاهش پیچیدگی: به جای مدیریت یک برنامه بزرگ، ماژولاریتی کد را به بخش‌های کوچک‌تر و ساده‌تر برای مدیریت تبدیل می‌کند.**

مزایای برنامه‌سازی ساخت‌یافته

۱- قابلیت خواندن و نگهداری کد:

- **کدهای ساخت‌یافته به دلیل سازمان‌دهی بهتر و استفاده از توابع و ماژول‌ها، خوانایی بیشتری دارند.**
- **کامنت‌گذاری مناسب و استفاده از نام‌گذاری واضح برای توابع و متغیرها باعث می‌شود تا برنامه‌نویسان دیگر بتوانند کد را به راحتی متوجه شوند.**

مزایای برنامه‌سازی ساخت‌یافته

۲. سهولت در اشکال‌زدایی و بهینه‌سازی:

- **به دلیل استفاده از توابع و ماژول‌ها، می‌توان بخش‌های خاصی از برنامه را به صورت مجزا تست و اشکال‌زدایی کرد.**
- **عیب‌یابی سریع‌تر: وقتی که خطایی در برنامه رخ می‌دهد، می‌توان آن را به راحتی به یک تابع یا ماژول خاص محدود کرد.**
- **بهینه‌سازی آسان‌تر: برنامه‌نویسان می‌توانند به راحتی توابع یا ماژول‌هایی که نیاز به بهینه‌سازی دارند را تشخیص دهند و روی همان بخش‌ها کار کنند.**

مزایای برنامه‌سازی ساخت‌یافته