



اصول و طراحی زبانها	
❖ تاریخچه مختصری از زبانهای برنامه سازی	
✓ زبان Fortran برای کارهای علمی و محاسباتی ساخته شده بود.	
YEAR	LANGUAGE
1950-55	Assembly
1956-60	Fortran , Algol 58 , Algol 60 , Cobol , Lisp
1961-65	FortranIR , Cobol 61 , Algol 61 , Snobol , APL
1966-70	Fortran 66 , Cobol 65 , Snobol 4 , Simula 67 , Basic , APL 360
1971-75	Pascal , Cobol 74 , APL (standard) , C
1976-80	Ada , Fortran 77 (standard)
1981-85	Prolog
1985-90	C++ , Fortran 90

اصول و طراحی زبانها	
❖ تاریخچه مختصری از زبانهای برنامه سازی	
✓ زبانهای محاسباتی (مبتنی بر اعداد)	
<ul style="list-style-type: none"> ▪ محاسبات با حجم زیاد را با دقت اعشار بالا انجام دهند. ▪ به انواع توابع ریاضی کتابخانه ای مجهز شده باشند. ▪ Fortran و Algol نمونه هایی از زبان های محاسباتی هستند. ▪ اهداف اصلی Algol در آن زمان شامل موارد زیر بود: <ul style="list-style-type: none"> • این زبان باید به ریاضیات استاندارد نزدیک باشد. • جهت توصیف الگوریتمها مناسب باشد. • برنامهها باید به زبان ماشین ترجمه گردند. • این زبان نباید وابسته به یک ماشین خاصی باشد. 	

فصل اول

اصول و طراحی زبانها

❖ تاریخچه مختصری از زبانهای برنامه سازی

✓ زبانهای تجاری

- برای کاربردهای تجاری، حسابداری، انبار و بورس
- تسریع در ورود اطلاعات، نگهداری سابقه و خروجی های واضح در قالب نمودار، جدول و ...
- هدف از زبان های تجاری این بود که، برنامه‌هایی که ایجاد می‌شوند از متن‌هایی به شکل متن انگلیسی استفاده کنند.
- Cobol، زبانی جهت کاربردهای تجاری می‌باشد که ارتش آمریکا برای نگهداری سوابق و اطلاعات نظامی خود از این زبان استفاده می‌کند.
- هدف مهم در Cobol، نوشتن برنامه ای است که به زبان انگلیسی نزدیک باشد. در این زبان پایان هر دستور با نقطه مشخص می‌شود.
- Cobol قابلیت خوانایی بالایی دارد اما نحو رسمی ندارد و برنامه نویسی در آن دشوار است.

فصل اول

اصول و طراحی زبانها

❖ تاریخچه مختصری از زبانهای برنامه سازی

✓ زبان هوش مصنوعی

- LISP، به عنوان یک زبان تابعی پردازش کننده لیست طراحی شد.
- LISP جهت کاربردهای هوش مصنوعی شامل ویژگی‌های زیر است:
 - یک زبان تابعی پردازش کننده لیست است.
 - عملیات جستجو را به خوبی انجام می‌دهد.
 - پیاده سازی بازیهای کامپیوتری در آن به خوبی صورت می‌گیرد.
 - قابلیت‌های مناسبی جهت پردازش متن دارد.
 - رشته‌هایی از نمادها می‌توانند توسط رشته‌های دیگر جایگزین شوند (تفسیر ماشین خودکار).
- نکته: LISP، جهت پردازش لیست همه منظوره طراحی شد ولی Prolog یک زبان تک منظوره است که ساختار آن بر اساس منطق ریاضی می‌باشد.

فصل اول

اصول و طراحی زبانها

❖ تاریخچه مختصری از زبانهای برنامه سازی

✓ زبانهای سیستمی

- برای نوشتن نرم افزارهای سیستمی نظیر سیستم عامل و پیاده سازی کامپایلرها و ... کاربرد دارند.
- باید قادر به دستیابی به سخت افزار و ایجاد ارتباط با آن باشند.
- اسمبلی، PL/I، C
- PL/I ویژگیهای عددی Fortran را با خصوصیات برنامه سازی تجاری Cobol ترکیب کرد
PL/I = Cobol + Fortran

فصل اول

اصول و طراحی زبانها

❖ تاثیر محیط اجرایی بر روی طراحی و پیاده سازی زبانها

- ✓ سخت افزاری که زبان را پشتیبانی می کند تاثیر زیادی بر طراحی زبان برنامه نویسی دارد.
- ✓ محیطی (شامل سیستم عامل و سخت افزار)، که برنامه در آن ایجاد، تست و اشکال زدایی می شود، محیط میزبان و محیطی که برنامه بر روی آن اجرا می شود محیط عملیاتی (مقصد) نام دارد.
- ✓ در اغلب موارد محیط میزبان و محیط مقصد یکسان است.
- ✓ گاهی اوقات ممکن است محیط عملیاتی با محیط میزبان متفاوت باشد.
- ✓ مثال: برنامه هایی که با پسوند jar جهت اجرا بر روی یک موبایل در یک کامپیوتر ایجاد می شوند که در این مورد محیط عملیاتی، محیط موبایل و محیط کامپیوتر محیط میزبان است.

فصل اول

اصول و طراحی زبانها

❖ تاثیر محیط اجرایی بر روی طراحی و پیاده سازی زبانها

✓ محیط دسته ای (Batch)

- کاربر با برنامه هیچ گونه تعامل و محاوره ای ندارد و ترتیب اجرای برنامه در خود برنامه گنجانده شده است.
- کاربر ورودی های مورد نیاز برنامه را وارد کرده و سپس منتظر می ماند تا اینکه برنامه پردازش شود و در نهایت خروجی بدست آید.
- فایل ها متداول ترین ابزار جهت ورودی/خروجی هستند.
- خطایی که اجرای برنامه را خاتمه دهد قابل اصلاح بوده ولی هزینه بر است، زیرا پس از پردازش و تصحیح خطا، برنامه مجدداً باید به طور کامل اجرا شود.
- زبان استفاده شده در محیط دسته ای محدوده زمانی مشخصی ندارد. زبانهایی مثل Fortran، Cobol و Pascal ابتدا برای محیطهای دسته ای طراحی شدند ولی امروزه در محیطهای محاوره یا سیستم تعبیه شده نیز به کار می روند.

فصل اول

اصول و طراحی زبانها

❖ تاثیر محیط اجرایی بر روی طراحی و پیاده سازی زبانها

✓ محیط محاوره ای (Interactive)

- برنامه مستقیماً با کاربر تعامل دارد.
- خروجی در نمایشگر نشان داده می شود.
- از سیستم اشتراک زمانی برای انجام کارهای مختلف، که در یک زمان واحد به کامپیوتر داده می شود استفاده می کنند.
- پردازش خطا از اهمیت کمتری برخوردار است زیرا کاربر به صورت online در صورت بروز خطا می تواند برنامه را دستکاری و تصحیح کند.
- C،C++ زبانهای برنامه های محاوره ای مناسب هستند.

فصل اول

اصول و طراحی زبانها

❖ تاثیر محیط اجرایی بر روی طراحی و پیاده سازی زبانها

✓ محیطهای سیستم تعبیه شده (Embedded system)

- سیستم کامپیوتری که جهت کنترل بخشی از یک سیستم بزرگ مثل هواپیما یا ماشین آلات صنعتی استفاده می‌شود، سیستم کامپیوتری تعبیه شده یا توکار گفته می‌شود.
- خرابی سیستم ممکن است زیانهای جدی به بار آورد.
- معمولاً به صورت بلادرنگ هستند یعنی در یک محدوده زمانی از قبل تعیین شده باید به ورودی‌ها پاسخ دهند.
- برنامه‌های این محیطها معمولاً بدون سیستم عامل، بدون سیستم فایل و بدون دستگاههای I/O اجرا می‌شوند. هر کدام از برنامه‌ها رویه‌های مخصوصی برای ارتباط با دستگاههای ورودی/ خروجی سیستم بزرگ دارند و به تعامل با آنها می‌پردازند.
- قابلیت اطمینان در این محیطها از اهمیت خاصی برخوردار است.
- اداره خطاها و استثناها باید به خوبی انجام گیرد. خاتمه برنامه به جز در موارد خرابی کلی سیستم، قابل قبول نیست و کاربری وجود ندارد که به صورت محاوره ای خطاها را برطرف سازد.
- اغلب از کامپیوترهای RISC که تعداد دستورات محدودی دارند استفاده می‌شود. زبانهای ++C و Ada.C برای نوشتن برنامه‌های مربوط به سیستم توکار مفید هستند.

فصل اول

اصول و طراحی زبانها

❖ تاثیر محیط اجرایی بر روی طراحی و پیاده سازی زبانها

✓ محیط کامپیوترهای شخصی

- کارایی، کمتر مد نظر قرار می‌گیرد.
- هدف اصلی راحتی کاربر و داشتن یک محیط گرافیکی ساده است.
- نمونه مشهور آن برنامه نویسی تحت محیط ویندوز است.
- برنامه نویسی شیء‌گرا، مدل مناسبی برای چنین محیط‌هایی است.
- زبان‌هایی مثل Java، ++C یا ++C در چنین محیطی کاربرد زیادی دارند.

فصل اول	اصول و طراحی زبانها
<p>❖ تاثیر محیط اجرایی بر روی طراحی و پیاده سازی زبانها</p> <p>✓ محیط شبکه و اینترنت</p> <ul style="list-style-type: none">▪ اینترنت اولیه، دو سرویس FTP و Telnet داشت.▪ بعد از ایجاد زبان HTML و پروتکل انتقال HTTP و استفاده از وب جهانی (WWW) نقش زبان برنامه سازی تغییر کرد.▪ این زبانها باید تعامل بین کامپیوترهای Client و Server را امکان پذیر سازند.	

فصل اول	اصول و طراحی زبانها
<p>❖ دامنه کاربرد زبانها</p> <p>✓ کاربردها در دهه ۱۹۶۰</p> <ul style="list-style-type: none">▪ پردازش تجاری از زبان Cobol▪ محاسبات علمی از Fortran و Algol▪ برنامه نویسی سیستمی از اسمبلر یا Forth▪ کاربردهای هوش مصنوعی از Snobol و Lisp <p>✓ کاربردهای قرن ۲۱</p> <ul style="list-style-type: none">▪ پردازش تجاری از زبان ++C, Java, Cobol▪ محاسبات علمی از ++C, Fortran, C, Java▪ برنامه نویسی سیستمی از ++C و Java▪ کاربردهای هوش مصنوعی Lisp و Prolog▪ انتشارات از Text و Postscript▪ کاربردهای جدید (مانند شی گراهاو...)	

فصل اول	اصول و طراحی زبانها
<p>❖ ویژگی‌های یک زبان خوب</p> <p>✓ وضوح، سادگی و یکپارچگی:</p> <ul style="list-style-type: none"> ▪ جامعیت مفهومی: مفاهیم و ابزارهای موجود در یک زبان و قوانین ترکیب آنها در یک زبان برنامه سازی ▪ نحو یک زبان روی نوشتن، تست کردن، اصلاح و درک زبان اثرگذار است. ▪ نحوی که مختصر و رمزگونه باشد برنامه نویسی را کوتاه تر و ساده تر می‌کند، ولی قابلیت خوانایی را کاهش می‌دهد (APL). ▪ خوانایی برنامه: تفاوت‌های معنایی منعکس کننده تفاوت‌های نحوی باشد. <p>✓ قابلیت تعامد: امکان ترکیب ویژگی‌های مختلف زبان و با معنا بودن ترکیب حاصل</p> <ul style="list-style-type: none"> ▪ مثال: ترکیب عبارت و ساختار شرطی (دستور محاسباتی در دستور if) ▪ مزیت: یادگیری زبان ساده و نوشتن برنامه راحت ▪ معایب: کامپایل بدون خطا در ترکیب‌هایی که منطق روشن و اجرای کارآمدی ندارند. 	

فصل اول	اصول و طراحی زبانها
<p>❖ ویژگی‌های یک زبان خوب</p> <p>✓ طبیعی بودن برای کاربردها</p> <ul style="list-style-type: none"> ▪ زبانها باید ساختمان داده، عملگرها، دستورات کنترلی و نحو مناسب برای مسئله ای که باید حل شود را داشته باشند. ▪ با توجه به کاربرد آن زبان، امکانات آن نیز موجود باشد. <p>✓ پشتیبانی از انتزاع (Abstraction)</p> <ul style="list-style-type: none"> ▪ زبان برنامه نویسی تعداد معینی نوع داده اولیه دارد. امکان دارد برنامه نویس یک نوع داده ای جدید (انتزاعی) را نیاز داشته باشد که در زبان برنامه نویسی وجود ندارد. ▪ در زبان‌هایی مانند C++ با استفاده از class و Ada با استفاده از Package برنامه نویسی می‌تواند نوع داده‌های انتزاعی (ADT) مورد نظر خود را تولید کند. ▪ استفاده از جنبه‌های انتزاعی، قابلیت اطمینان را افزایش خواهد داد. 	

فصل اول	اصول و طراحی زبانها
❖ ویژگی‌های یک زبان خوب	
✓ سهولت در بازرسی برنامه	
<ul style="list-style-type: none"> ▪ برنامه‌هایی که ساختار نحوی و معنایی ساده تری دارند بازرسی آنها نیز ساده تر می باشد. ▪ بررسی درستی برنامه به دو صورت رسمی مانند روش‌های ریاضی و غیر رسمی نظیر تست برنامه با ورودی‌های مختلف انجام می‌شود. 	
✓ محیط برنامه نویسی	
<ul style="list-style-type: none"> ▪ وجود ویراستارهای خاص، امکانات نگهداری و اصلاح نسخه های متفاوت ▪ ایجاد برنامه‌ها و عیب یابی ساده تر با محیط برنامه نویسی قوی 	
✓ قابلیت حمل برنامه	
<ul style="list-style-type: none"> ▪ یک زبان خوب باید بتواند بر روی سیستم‌های مختلف کامپیوتری به سادگی انتقال یافته و اجرا شود. ▪ زبان جاوا این ویژگی را در حد بسیار بالایی دارد. ▪ با افزایش قابلیت حمل، انعطاف پذیری و کارآیی برنامه کاهش می یابد. 	

فصل اول	اصول و طراحی زبانها
❖ ویژگی‌های یک زبان خوب	
✓ ۳ معیار اصلی هزینه:	
<ul style="list-style-type: none"> ▪ هزینه اجرای برنامه: زمان اجرای یک برنامه خصوصاً برنامه‌های بزرگی که به دفعات زیادی اجرا می شوند. بستگی به کامپایلر دارد ولی امروزه زیاد مهم نیست. هزینه اجرای برنامه شامل استفاده از منابع کامپیوتری خصوصاً CPU, RAM, Disk می باشد. ▪ هزینه ترجمه برنامه: مدت زمانی که طول می کشد تا برنامه کامپایل شود برنامه های دانشجویی چندین بار ترجمه شده و کمتر اجرا می شوند. هزینه ترجمه برنامه شامل استفاده از منابع کامپیوتری خصوصاً CPU, RAM, Disk می باشد. ▪ هزینه نگهداری برنامه: هزینه تطبیق برنامه با جوانب محیطی، ایجاد امکانات جدید در نرم افزار و برطرف کردن اشکالات، هزینه ترمیم خطا بعد از اجرا، توسعه و تغییر سیستم عامل 	

فصل اول	اصول و طراحی زبانها
❖ نحو (syntax) و معنای (semantic) زبان	
✓ نحو زبان برنامه سازی، ظاهر آن زبان است.	
✓ قواعد نحوی مشخص می کنند که دستورات، اعلانها و سایر ساختارهای زبان چگونه نوشته می شوند	
✓ معنای زبان همان مفهومی است که به ساختارهای نحوی زبان داده می شود.	
✓ نحو: اعلان آرایه بردار عنصری از نوع صحیح :	
v:array [0..9] of integer;	Pascal
int v[10];	C
✓ معنا: هر دو اعلان بدین معناست که ۱۰ خانه برای اعداد صحیح در نظر گرفته شود.	

فصل اول	اصول و طراحی زبانها
❖ مدل‌های محاسباتی زبان	
✓ زبانهای دستوری (imperative) یا رویه ای: زبانهای مبتنی بر فرمان یا دستورگرا (شامل دنباله ای از دستورات است و اجرای هر دستور موجب می شود مترجم مقدار یک یا چند خانه حافظه را تغییر دهد)	
▪ مانند C, C++, Algol, Fortran, Cobol و پاسکال و ...	
✓ زبانهای تابعی (applicative): به جای مشاهده تغییر حالت ماشین، عملکرد برنامه دنبال می شود.	
▪ مانند ML و LISP $function_n(...(function_2(function_1(data)) ...)$	
✓ زبانهای قانونمند (rule-based): شرایطی را بررسی می کنند و در صورت برقرار بودن آنها فعالیت را انجام می دهند. (هوش مصنوعی و سیستم‌های خبره)	
▪ مانند Prolog $enable\ condition_1 \rightarrow action_1$	
✓ برنامه نویسی شی گرا (object-oriented): اشیای پیچیده به عنوان بسطی از اشیای ساده ساخته می شوند و خواصی را از اشیای ساده به ارث می برند.	
▪ مانند C++, Java, Smalltalk	

فصل اول

اصول و طراحی زبانها

❖ استاندارد سازی زبانها

✓ برای پی بردن به معنای دستور و خروجی آن سه راه حل وجود دارد:

- به مستندات زبان مراجعه شود.
- برنامه را در کامپیوتر تایپ و اجرا کنید.
- به استاندارد زبان مراجعه شود.

✓ استانداردهای زبان دو دسته اند:

- استاندارد خصوصی: توسط شرکت سازنده یا مالک زبان ارائه می شوند.
- استاندارد عمومی: توسط سازمانهای معروف مثل ISO و ANSI ارائه می شوند.

✓ مسائل مهم در استفاده موثر از استاندارد:

- زمان سنجی: چه زمانی باید زبان استاندارد شود؟
- اطاعت و پیروی: کامپایلر پیرو یک زبان، فقط برنامه های نوشته شده مطابق با استاندارد آن زبان را ترجمه می کند و خروجی های مناسب تولید می کند.
- کهنگی و منسوخ شدن: کی استاندارد کهنه می شود و چگونه باید آن را اصلاح کرد؟

فصل اول

اصول و طراحی زبانها

❖ محیط های برنامه نویسی

✓ الف) محیط برنامه نویسی در دو زمینه بر طراحی زبان تاثیر گذاشته است:

- کامپایل کردن مجزای زیربرنامه و سایر بخشهای برنامه

• کامپایلر باید این اطلاعات را داشته باشد:

○ مشخه تعداد، ترتیب و نوع پارامترهای زیربرنامه

○ اعلان نوع داده

○ تعریف نوع داده

- تست و اشکال زدایی

• مانند ویژگیهای ردیابی اجرا، نقاط کنترلی

فصل اول

اصول و طراحی زبانها

❖ محیط های برنامه نویسی

✓ ب) محیط های کاری

▪ محیط کاری، خدماتی مثل ذخیره داده ها، رابط گرافیکی کاربر، امنیت و خدمات ارتباطی را فراهم می کند.

✓ ج) زبانهای کنترل کار و فرآیند

▪ مفهوم کنترل کار به چارچوبهای محیط برمی گردد.

▪ کاربر کنترل مستقیم بر روی مراحل مختلف برنامه دارد.

فصل دوم

اثرات معماری ماشین

❖ مقدمه

✓ در توسعه یک زبان برنامه نویسی سه عامل بر روی طراحی زبان موثر است:

▪ کامپیوتری که برنامه بر روی آن اجرا می شود.

▪ مدل اجرا یا کامپیوتر مجازی که آن زبان را روی سخت افزار اجرا می کند.

▪ مدل محاسباتی که زبان آن را پیاده سازی می کند.

✓ کامپیوتر ها می توانند در یکی از سه شکل زیر باشند:

▪ کامپیوتر واقعی (یا سخت افزاری)

▪ کامپیوتر شبیه سازی شده نرم افزاری (میان افزار)

▪ کامپیوتر مجازی که ترکیبی از سخت افزار و نرم افزار است

▪ ترکیبی از روشهای بالا