# What is a product?

A product is any item or service you sell to serve a customer’s need or want. They can be physical or virtual. Physical products include durable goods (like cars, furniture, and computers) and nondurable goods (like food and beverages). Virtual products are offerings of services or experiences (such as education, software, and streaming services). A product may also be a hybrid — including both physical and virtual elements (like a kitchen appliance with its own mobile app). Hybrid products are becoming more common as traditionally analog products incorporate [digital technology](https://www.aha.io/roadmapping/guide/enterprise-transformation/what-is-digital-transformation) as a way to better reach and serve customers.

# These categories can help us understand different types of products. But they are not hard and fast rules. For example, software can be considered both a product and a service. Traditionally, you would purchase a physical version of the program and install it on your computer – but today most software products are sold virtually. This approach uses a web – based delivery model and customers pay a recurring subscription fee to access the software. This is how the term software -as-a-service (SaaS) originated.

**Characteristics of a product**

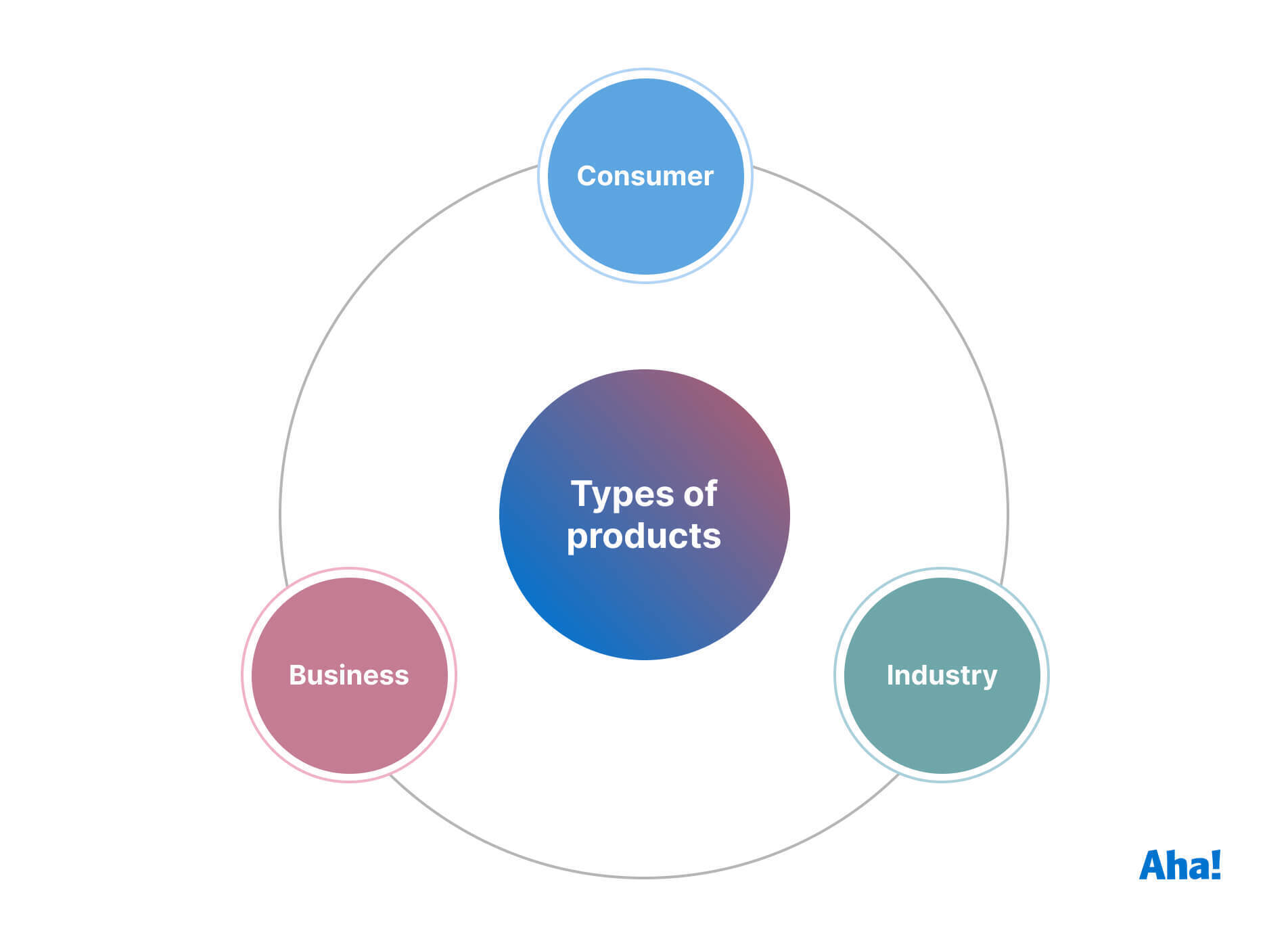
A live concert, shampoo, and a fitness app. These are all examples of products but the similarities between them are few. This is part of what makes [product development](https://www.aha.io/roadmapping/guide/what-is-product-development) so exciting — endless [possibilities](https://www.aha.io/blog/5-major-trends-shaping-the-future-of-product-development) exist to create new offerings for all kinds of customers.

Though product characteristics vary widely, a few basic ones are universal. A product is:

* **Intended for customers.** This differentiates products from projects or anything else you may produce for your own use or enjoyment. Products are typically created with the intent of being sold and consumed by someone else — whether that is an individual consumer or a business.
* **Created to provide benefits to a market.**Identifying a market need and meeting it can be challenging — but at a base-level, a product should provide some sort of advantage to users.
* **Exchanged for value.**The most typical value exchange is money — meaning, products have a price and can be bought and sold. In some cases products will be offered in exchange for feedback, exposure, a trade, or other forms of [value](https://www.aha.io/roadmapping/guide/product-strategy/how-to-measure-product-value).

**Types of products**

Beyond physical, virtual, and hybrid, products can be classified in other ways. You can start by splitting products among three major customer categories — consumer, business, and industry.



**Consumer products**

Consumer products, or business-to-consumer (B2C) products, are sold to end-users and intended for personal use. The consumer product category is commonly broken down further by purchasing behavior — as different characteristics can influence the way customers buy products. The table below explains the four major consumer product types by purchasing behavior:

|  |  |
| --- | --- |
| **Purchasing behavior** | **Description** |
| **Convenience** | Convenience products are purchased frequently and with little planning or effort. They are widely available, easy to obtain, and typically have a low price.  *Example: Magazines, on-demand software and services* |
| **Shopping** | Shopping products are purchased less frequently than convenience products and have a higher price. Buyers compare attributes such as quality, style, and price before making a purchasing decision.  *Example: Clothing brands, airline tickets* |
| **Specialty** | Specialty or niche products have features that appeal to a specific group of customers. This type of product requires more targeted promotion to reach the right people.  *Example: Vertical market software such as real-estate or banking applications* |
| **Unsought** | Unsought products have little awareness or proactive demand among customers. Because customers do not perceive an immediate need for these products, the benefits must be directly promoted to generate interest.  *Example: Life insurance, reference books* |

**Business products**

Business products, or business-to-business (B2B) products, help other companies create their own products or operate their business. Business products can also be referred to as horizontal market products — present in multiple industries and supporting a wide range of business needs. Examples of business products include raw materials, equipment, supplies, business services, and software.

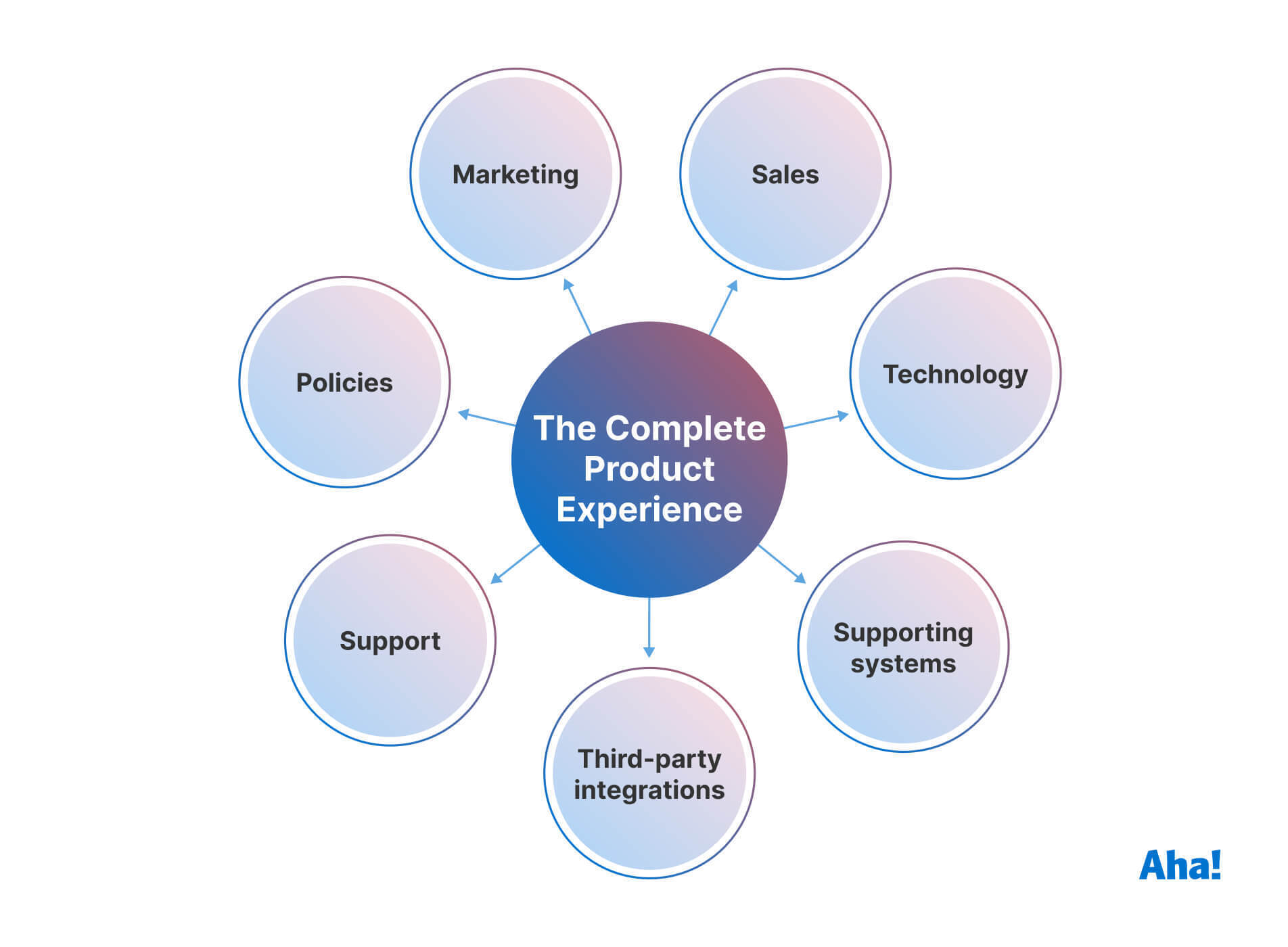
Business software is a major sub-category of B2B products. Examples include accounting, customer relationship management (CRM), human resource management, and [product development software](https://www.aha.io/roadmapping/guide/what-is-product-development-software). These applications can be further categorized by the size of the company that uses them (e.g., enterprise software).

**Industry products**

Industry products, or vertical market products, serve broad business sectors such as energy, healthcare, financial services, or information technology. Rather than cater to a large variety of use cases, industry products are tailored to meet the needs of a specific industry (e.g., a healthcare application for managing patient data).

**Understanding the Complete Product Experience (CPE)**

A product is more than just the item or service that is sold. Of course, the [features and benefits](https://www.aha.io/roadmapping/guide/requirements-management/what-are-product-features) a product provides are important — but real [product value](https://www.aha.io/roadmapping/guide/product-strategy/what-is-product-value) is all the work you do to [strategize](https://www.aha.io/roadmapping/guide/product-management/what-is-a-product), build, launch, market, sell, and support an offering. For product builders, this means you need to consider and optimize every way a customer interacts with your product and company.



This concept is called the [Complete Product Experience](https://www.aha.io/roadmapping/guide/product-strategy/complete-product-experience) (CPE). Brian de Haaff, co-founder and CEO, defines the CPE in [*Lovability*](https://www.aha.io/lovability) — the bestselling book for company and product builders. The following is an excerpt from *Lovability*:

*1. Marketing is how potential customers learn about your product and determine if it might be a fit to help them solve their problem. This is taking on new forms as people grow increasingly connected: social platforms, online reviews, and company-published content.*

*2. Sales is the process of prospects learning more about the product from a company representative and possibly using it in a trial. They educate themselves about the product and get the information they need to determine if the solution is right for them.*

*3. Technology refers to the core set of features that customers pay for. In our case, that is the online software that they log into our servers to use. For others, it could be the actual phone, credit card, or even an insurance policy that is purchased. However, technology should represent not the end of a transaction but the beginning of a transparent, interactive relationship.*

*4. Supporting systems make it possible to deliver the product. These are internal systems that the customer rarely sees but which can have a huge impact on their overall happiness: billing, provisioning, analytics, and more. For example, if you call customer support and the representative always seems to have a comprehensive history of your purchases and support issues at their fingertips, you can thank supporting systems.*

*5. Third-party integrations enable new products to fit into how the customer already lives and works. All products exist in an ecosystem, so they have to play nice with the other products the customer is already using and the way that customer already works.*

*6. Support is everything from answering customer questions to training and even helping customers integrate your product with their existing systems. Support describes all activity that helps the customer achieve something meaningful with a product.*

*7. Policies are the rules that companies set to govern how they do business. At their best, they provide a framework for employees to be their best. At their worst, they create unresolvable frustration that drives customers to ask “to speak with the manager.”*

Ultimately, the product is the summation of all touchpoints that form the [relationship between your company and the customer](https://www.aha.io/blog/understanding-the-heart-of-the-customer). It is important to consider how the [components](https://www.aha.io/blog/7-components-of-your-complete-product-experience) described above interact with each other. Creating a seamless experience requires bringing together product, marketing, sales, and support teams to [optimize each step of the customer’s journey](https://www.aha.io/roadmapping/guide/enterprise-transformation/what-is-customer-experience) and create lasting joy.

**Read more:**[What is the Complete Product Experience?](https://www.aha.io/roadmapping/guide/product-strategy/complete-product-experience)

**How to launch a new product**

Product launches are exciting. But they are only one piece of [product development](https://www.aha.io/roadmapping/guide/what-is-product-development) — the massive body of work encompassing everything it takes to bring a new product to life. This includes the "active" building and launch activities as well as the behind-the-scenes strategizing, planning, and analysis.

For most businesses, there are [seven stages](https://www.aha.io/roadmapping/guide/stages-of-product-development) of developing a new product.

|  |  |
| --- | --- |
| **1. Strategize** | Define [goals and initiatives](https://www.aha.io/roadmapping/guide/product-strategy/what-are-product-goals-and-initiatives) |
| **2. Ideate** | Brainstorm and capture promising ideas |
| **3. Plan** | Refine ideas based on strategic goals, estimated product value, and [team capacity](https://www.aha.io/roadmapping/guide/release-management/capacity-planning) |
| **4. Showcase** | [Share roadmaps](https://www.aha.io/roadmapping/guide/roadmap/ultimate-guide) and go-to-market plans with stakeholders |
| **5. Build** | Deliver new functionality via [agile development](https://www.aha.io/roadmapping/guide/agile/agile-software-development) |
| **6. Launch** | Bring those exciting new capabilities to market |
| **7. Analyze** | Assess realized product value by tracking customer usage and [love](https://www.aha.io/blog/the-top-10-most-loved-products-have-this-in-common) |

It requires a concerted effort throughout every stage of the process to successfully deliver a new product. When you are approaching the launch phase, a[product launch checklist](https://www.aha.io/roadmapping/guide/release-management/what-is-a-good-product-launch-checklist) can help make sure everything goes according to plan.

**GRAPHIC DESIGN**

Graphic design is a profession, academic discipline and applied art whose activity consists in projecting visual communications intended to transmit specific messages to social groups, with specific objectives. Graphic design is an interdisciplinary branch of design and of the fine arts. Its practice involves creativity, innovation and lateral thinking using manual or digital tools, where it is usual to use text and graphics to communicate visually. Due to its interdisciplinary nature, graphic design can be performed in different areas of application: branding, technical and artistic drawing, signage, photography, image and video editing, 3D modeling, animation, programming, among other fields. The role of the graphic designer in the communication process is that of encoder or interpreter of the message. They work on the interpretation, ordering, and presentation of visual messages. Usually, graphic design uses the aesthetics of typography and the compositional arrangement of the text, ornamentation, and imagery to convey ideas, feelings, and attitudes beyond what language alone expresses. The design work can be based on a customer’s demand, a demand that ends up being established linguistically, either orally or in writing, that is, that graphic design transforms a linguistic message into a graphic manifestation.

Graphic design has, a field of application, different areas of knowledge focused on any visual communication system. For example, it can be applied in advertising strategies, or it can also be applied the aviation world or space exploration. In this sense, in some countries graphic design is related as only associated with the production of sketches and drawings, this is incorrect, since visual communication is a small part of a huge range of types and classes where it can be applied.

With origins in Antiquity and the Middle Ages, graphic design as applied art initially linked to the boom of rise of printing in Europe in the 15th century and the growth of consumer culture in the Industrial Revolution. From there it emerged as a distinct profession in the West, closely associated with advertising in the 19th century and its evolution allowed its consolidation in the 20th century. Given the rapid and massive growth in information exchange today, the demand for experience designers is greater than ever, particularly because of the development of new technologies and the need to pay attention to human factors beyond the competence of the engineers who develop them.

**MEANING OF THE ABBREIVATIONS**

1. Hppt: High Pressure Production Trap
2. Hppt: Humane Pressure Point Tactics
3. Hppt: Honors Program in Physical
4. Hppt: High-Power Physics Testing
5. Hppt: Hyper Text Transfer Protocol
6. www: World wide web
7. UPI: Unified Payments Interface
8. IP: Internet Protocol

b. Intellectual Property

5. HTTP: Hypertext Transfer Protocol

6. TCP: Transmission Control Protocol

**Website**

A **site** or **website** is a central location of [web pages](https://www.computerhope.com/jargon/w/webpage.htm) that are related and accessed by visiting the [home page](https://www.computerhope.com/jargon/h/homepage.htm) of the website using a [browser](https://www.computerhope.com/jargon/b/browser.htm). For example, the Computer Hope website address [URL](https://www.computerhope.com/jargon/u/url.htm) (Uniform Resource Locator) is [https://www.computerhope.com](https://www.computerhope.com/). From our [home page](https://www.computerhope.com/jargon/h/homepage.htm), you could access any of the web pages (like this one) on our website. The image shows how the Computer Hope website looked in 2021.

**How to open a website**

To view a website requires a [browser](https://www.computerhope.com/jargon/b/browser.htm) (e.g., [Internet Explorer](https://www.computerhope.com/jargon/m/msie.htm), [Edge](https://www.computerhope.com/jargon/m/microsoft-edge.htm), [Safari](https://www.computerhope.com/jargon/s/safari.htm), [Firefox](https://www.computerhope.com/jargon/f/firefox.htm), or [Chrome](https://www.computerhope.com/jargon/c/chrome.htm)). For example, you are reading this web page using a browser. Once in a browser, you can open a website by entering the [URL](https://www.computerhope.com/jargon/u/url.htm) in the [address bar](https://www.computerhope.com/jargon/a/addrebar.htm). For example, typing "https://www.computerhope.com" opens the Computer Hope home page. If you don't know the URL of the website you want to visit, you can use a [search engine](https://www.computerhope.com/jargon/s/searengi.htm) to find the website on the Internet.

**What is the difference between a website and a web page?**

A website refers to a central location with more than one web page or several web pages. For example, [Computer Hope](https://www.computerhope.com/) is considered a website, which contains thousands of different web pages, including the page you're reading now.

**Types of websites**

There are billions of websites on the Internet today that can be broken into one of the following types of website categories. Keep in mind that it is possible for a website to fall into more than one of the following categories. For example, a website may also be a [forum](https://www.computerhope.com/jargon/f/forum.htm), [webmail](https://www.computerhope.com/jargon/w/webmail.htm), [blog](https://www.computerhope.com/jargon/w/weblog.htm), or [search engine](https://www.computerhope.com/jargon/s/searengi.htm).

* [Archive website.](https://www.computerhope.com/jargon/w/website.htm#archive)
* [Blog (weblog).](https://www.computerhope.com/jargon/w/website.htm#blog)
* [Business website and corporate website.](https://www.computerhope.com/jargon/w/website.htm#business)
* [Community website.](https://www.computerhope.com/jargon/w/website.htm#community)
* [Content website and information website.](https://www.computerhope.com/jargon/w/website.htm#content)
* [Dating website.](https://www.computerhope.com/jargon/w/website.htm#dating)
* [Dynamic website.](https://www.computerhope.com/jargon/w/website.htm#dynamic)
* [E-commerce website.](https://www.computerhope.com/jargon/w/website.htm#e-commerce)
* [Educational website.](https://www.computerhope.com/jargon/w/website.htm#educational)
* [Gaming website.](https://www.computerhope.com/jargon/w/website.htm#gaming)
* [Government website.](https://www.computerhope.com/jargon/w/website.htm#government)
* [Help and Q&A website.](https://www.computerhope.com/jargon/w/website.htm#help)
* [Malicious website.](https://www.computerhope.com/jargon/w/website.htm#malicious)
* [Media sharing website.](https://www.computerhope.com/jargon/w/website.htm#media)
* [Mirror website.](https://www.computerhope.com/jargon/w/website.htm#mirror)
* [News website.](https://www.computerhope.com/jargon/w/website.htm#news)
* [P2P website and Torrent website.](https://www.computerhope.com/jargon/w/website.htm#p2p)
* [Personal website.](https://www.computerhope.com/jargon/w/website.htm#personal)
* [Personality website.](https://www.computerhope.com/jargon/w/website.htm#personality)
* [Portal](https://www.computerhope.com/jargon/w/website.htm#portal)
* [Review website.](https://www.computerhope.com/jargon/w/website.htm#review)
* [School website.](https://www.computerhope.com/jargon/w/website.htm#school)
* [Scraper website.](https://www.computerhope.com/jargon/w/website.htm#scraper)
* [Search engine website.](https://www.computerhope.com/jargon/w/website.htm#search)
* [Secure website](https://www.computerhope.com/jargon/w/website.htm#secure)
* [Social networking website.](https://www.computerhope.com/jargon/w/website.htm#social-networking)
* [Social news website.](https://www.computerhope.com/jargon/w/website.htm#social)
* [Static website](https://www.computerhope.com/jargon/w/website.htm#static)
* [Unsecure website](https://www.computerhope.com/jargon/w/website.htm#unsecure)
* [Webcomic website.](https://www.computerhope.com/jargon/w/website.htm#webcomic)
* [Webmail website.](https://www.computerhope.com/jargon/w/website.htm#webmail)
* [Wiki website.](https://www.computerhope.com/jargon/w/website.htm#wiki)

**Archive website**

An **archive website** is a site that keeps a record of the contents of one or more other websites. The [Internet Archive](https://www.computerhope.com/jargon/a/archive.htm) is the best example of an archive website.

**Blog (weblog)**

A **blog** is a website that is often created by an individual to keep a list of entries that interests them. See our [weblog](https://www.computerhope.com/jargon/w/weblog.htm) definition for a full description, services used to create a blog, and related pages. A **microblog website** is also another popular form of blogging website that limits the number of characters someone can post in each blog entry. [Twitter](https://www.computerhope.com/jargon/t/twitter.htm) is an example of a social networking website for a [microblog](https://www.computerhope.com/jargon/m/microblo.htm).

**Business website and corporate website**

A **business website** or **corporate website** is created to provide account information and access to customers, partners, clients, and potential customers.

**Community website**

A community website is a website or section of a website that helps bring the visitors visiting the site together using [chat](https://www.computerhope.com/jargon/c/chat.htm), [forums](https://www.computerhope.com/jargon/f/forum.htm), or another form of bulletin boards.

**Content website and information website**

A **content website** and **information website** are created with the intention of displaying unique [content](https://www.computerhope.com/jargon/c/content.htm) that is often related to a specific category. For example, Computer Hope could be considered a content site with computer-related content. Other categories could include a **political website** with content relating to politics or a political view, or a **religious website** with information about a specific religion.

**Dating website**

A **dating website** is a site set up to help connect people who may be interested in meeting other people or dating other people. Most dating websites require a small fee, require a description of yourself, and often asks questions to help find people that would best match your interests.

**Dynamic website**

A **dynamic website** is a website that uses a [database](https://www.computerhope.com/jargon/d/database.htm) and [server-side scripting](https://www.computerhope.com/jargon/s/server-side-scripting.htm) to help dynamically generate the content on the website. [WordPress](https://www.computerhope.com/jargon/w/wordpress.htm) sites and other [CMS](https://www.computerhope.com/jargon/c/cms.htm) (content management system) solutions make it easier for anyone to have a website without knowing a lot about HTML or programming. See our [dynamic website](https://www.computerhope.com/jargon/d/dynasite.htm) page for further information on dynamic and static websites.

An **e-commerce (electronic commerce) website** is any site that was created with the intention of selling online goods or services. [Amazon](https://www.computerhope.com/comp/amazon.htm) is an example of an e-commerce website. An e-commerce website may be broken down even further into one of the following subcategories.

An **affiliate website** is a website created with the intention of selling third-party products. For example, Amazon has an [affiliate program](https://www.computerhope.com/jargon/a/affiliat.htm) for anyone to link to their site and make a commission when products are purchased. An affiliate website should not be confused with an e-commerce website.

An **auction website** is a website that allows other people to sell their goods or services. For example, [eBay](https://www.computerhope.com/comp/ebay.htm) is one of the most well-known online auction websites. See our [online auction](https://www.computerhope.com/jargon/o/oauction.htm) page for further information and examples.

A **classified ads website** is a site that allows anyone to list goods or services, usually for free or at a small cost. [Craigslist](https://www.computerhope.com/jargon/c/craigslist.htm) is an example of a classified ads website.

A **crowdfunding website** is set up to help support a business, person, or another cause by making a one time or monthly payment. An example of a crowdfunding website is [Kickstarter](https://www.computerhope.com/jargon/k/kickstarter.htm).

**Educational website**

An educational website is any website containing content that aids in learning. It could be a [school website](https://www.computerhope.com/jargon/w/website.htm#school), a library website, a homework assistance website, or any other tools for gaining knowledge online.

**Gaming website**

A **gaming website** is any website that features games that can be played on the website. Often these online games are created using [HTML5](https://www.computerhope.com/jargon/h/html.htm), [Flash](https://www.computerhope.com/jargon/f/flash.htm), or [Java](https://www.computerhope.com/jargon/j/java.htm). Gaming websites should not be confused with a gaming content website with content relevant to gaming with no actual games to play on the website.

**Government website**

A **government website** is a department, local, or state government site that was created to help inform the public about government business and services. A local government website may also be set up to help promote tourism.

**Help and Q&A website**

A **help website** and **questions and answers website** is a site where anyone can post questions and other users help answer those questions. A full listing of help websites where you can ask any question is on the link below.

**Malicious website**

A **malicious website** is any website that was set up with the intention of [infecting](https://www.computerhope.com/jargon/i/infect.htm) another computer or collecting personal data. For example, a **malware website** is a site created with the intention of infecting any visitor with [malware](https://www.computerhope.com/jargon/m/malware.htm), [spyware](https://www.computerhope.com/jargon/s/spyware.htm), or a [trojan horse](https://www.computerhope.com/jargon/t/trojhors.htm). These types of sites could have a [download](https://www.computerhope.com/jargon/d/download.htm) that's infected and, if downloaded, infects your computer.

Other common malicious websites include **phishing websites**. These sites are designed to look like other official sites (e.g., your bank) with the hope that they can [phish](https://www.computerhope.com/jargon/p/phishing.htm) sensitive information such as your username and password.

**Fake news websites** are another type of malicious site created to appear as a legitimate source of news with the intention of helping to spread fear and lies.

**Media sharing website**

A **media sharing website** is any website that specializes in allowing visitors to share one or more types of [media](https://www.computerhope.com/jargon/m/media.htm). For example, [YouTube](https://www.computerhope.com/jargon/y/youtube.htm) is a site for sharing video media. [SoundCloud](https://www.computerhope.com/jargon/s/soundcloud.htm) is a site for sharing music. [Flickr](https://www.computerhope.com/jargon/f/flickr.htm) is a place to share photos. DeviantArt is a page for sharing art.

**Mirror website**

A **mirror website** is a complete duplicate of another website that is used when a website becomes overloaded. It helps with a website's speed in different parts of the world. See our [mirror](https://www.computerhope.com/jargon/m/mirrorin.htm) definition for further information. Also, although similar, a mirror site should not be confused with a scraper website or a [CDN](https://www.computerhope.com/jargon/c/cdn.htm).

**News website**

A **news website** is a site dedicated to giving the latest local or world news. A news site may also be dedicated to a specific topic. For example, many [computer-related news websites](https://www.computerhope.com/network/news.htm) are dedicated to talking about the latest computer and technology related news.

**P2P website and Torrent website**

A **P2P website** and **Torrent website** are sites created to list available [torrents](https://www.computerhope.com/jargon/b/bittoren.htm) that can be downloaded using a file-sharing program.

A **warez website** is similar to a torrent website, except that it stores and hosts music, video, and software that anyone can download to their computer. When referring to a [warez](https://www.computerhope.com/jargon/w/warez.htm) website, it is describing a site where [copyrighted](https://www.computerhope.com/jargon/c/copyrigh.htm) material is illegally downloaded.

**Personal website**

A **personal website** is a site created by an individual that talks about their personal life, family, life experiences, and maybe contains a [résumé](https://www.computerhope.com/jargon/r/resume.htm). Today, many people are creating personal websites as a blog or using a social networking website as a place to store information about themselves.

**Personality website**

A **personality website** is any website covering an individual, such as an artist, celebrity, musician, author, or any other person. These types of websites are set up by someone affiliated with the person, such as a publicist, agency, or fan of the personality.

**Portal**

A portal is an idea of a website or service that offers a broad range of services, such as e-mail, games, quotes, search, news, and stocks. See our [portal](https://www.computerhope.com/jargon/p/portals.htm) term page for further information.

**Review website**

A **review website** is any site that focuses on [reviews](https://www.computerhope.com/jargon/r/review.htm) about a product or service. For example, [Yelp](https://www.computerhope.com/jargon/y/yelp.htm) allows consumers to review businesses in their area. Other review sites may review other things such as movies or products. Also, many e-commerce sites have reviews from people who've purchased the product they're selling.

**Secure website**

A **secure site** or **secure website** is any website that transmits data securely ([encrypted](https://www.computerhope.com/jargon/e/encrypt.htm)). Secure websites are identified by looking for a lock next to the [URL](https://www.computerhope.com/jargon/u/url.htm) or a URL starting with "[https](https://www.computerhope.com/jargon/h/http.htm)" instead of "http."

**School website**

A **school website** is a site created to represent a local school or college. School sites have an overview of a school and give students and parents a place to log in and review grades and other school-related information.

**Scraper website**

A **scraper website** is a website that is illegally stealing (scraping) another website's content. Doing this could allow the person to generate advertising revenue if enough traffic was driven to the scraper website. However, these types of sites are against all advertisers' [TOS](https://www.computerhope.com/jargon/t/tos.htm) (terms of service) and, when caught, they are blocked by the company providing the advertising. A scraper website should not be confused with a mirror website that was set up with permission.

**Search engine website**

A **search engine website** is a website dedicated to helping people find information on the Internet. [Google](https://www.computerhope.com/jargon/g/google.htm) is an example of a search engine website. See our search engine definition for further information on [search engines](https://www.computerhope.com/jargon/s/searengi.htm) and related links.

**Social networking website**

A **social networking website** connects users with friends, family, celebrities, groups, and organizations. The service is usually free, on the condition that the website may collect and use the users information. [Facebook](https://www.computerhope.com/jargon/f/facebook.htm) and [Twitter](https://www.computerhope.com/jargon/t/twitter.htm) are examples of a social networking website. See our [social networking](https://www.computerhope.com/jargon/s/socinetw.htm) definition for other examples of social networks and further information.

**Social news website**

A **social news website** is a site that generates its content from its members and, once posted, all other members can vote if they enjoyed or liked the story. [Reddit](https://www.computerhope.com/jargon/r/reddit.htm) is an example of a social news website that allows everyone to post almost anything. It is a great place to find things that would most interest you on the Internet.

**Static website**

A **static website** is a website designed using only HTML and doesn't change or update automatically. See our [dynamic website](https://www.computerhope.com/jargon/d/dynasite.htm) page for further information on dynamic and static websites.

**Unsecure website**

A **unsecure site** or **unsecure website** is any website that does not transmit data securely ([encrypted](https://www.computerhope.com/jargon/e/encrypt.htm)). Unsecure websites are identified by looking for a URL starting with "[http](https://www.computerhope.com/jargon/h/http.htm)" instead of "https." When on an unsecure website, be cautious with transmitted information that could be intercepted and read with a [man-in-the-middle attack](https://www.computerhope.com/jargon/m/mitma.htm).

**Webcomic website**

A **webcomic website** is a site that posts a comic strip daily, weekly, or monthly.

**Webmail website**

A **webmail website** is a website that a person to view, send, and receive e-mail without the need for software. An example of a webmail program is [Gmail](https://www.computerhope.com/jargon/g/gmail.htm). See our [webmail](https://www.computerhope.com/jargon/w/webmail.htm) definition for other examples, information, and related links.

**Wiki website**

A **wiki website** is a site that is created using [Wiki](https://www.computerhope.com/jargon/w/wiki.htm) software, and is often edited and updated by more than one person.

**FIFTEEN CATEGORIES OF A WEBSITE**

## Categories of Websites

There are many kinds of websites on the internet. Below we’ve listed 15 of the most common ones and their uses.

### eCommerce

### Business

### Entertainment

### Nonprofit

### Web Portal

### Portfolio

### Blogs

### Educational

### Informational

### Social Media

### Brochures

### Events

### Online Forums

### Membership

1. Personal

# What is Assembly Language?

## Introduction to Assembly Language

Assembly Language is a low-level programming language. It helps in understanding the programming language of machine code. Machine code is a series of instructions that provide the necessary information to a user’s CPU (Central Processing Unit) to carry out a particular task (add, subtract, compare values, etc.). In computers, there is an assembler that helps in converting the assembly code into machine code executable. Symbolic machine code aims to understand the instruction and provide it to machine language for further processing. It mainly depends on the system architecture, whether it is an operating system or computer architecture.

Symbolic machine code mainly consists of mnemonic processor instructions or data and other statements or instructions. It establishes with the help of compiling high-level language source code like C and C++. It helps in fine-tuning the program.

### Key Highlights

* It is a low-level, human-readable, programmable language in which each ASM instruction relates to a computer machine code instruction.
* In comparison to other high-level languages, symbolic machine code refers to consistency. It only has a few operations, but it helps to know the algorithms and other control flows. It simplifies the code and makes it easier to debug.
* By learning the assembler language, the programmer can write the code to access registers and retrieve the memory address of pointers and values.
* If an individual wants to know how the system works and the processor as well, then symbolic machine code is the one that solves the purpose. It helps in all aspects, from understanding the program’s algorithm to the processor working and registering the computer’s register.

### History of Assembly Language

It was featured in 1949, and its history interconnects with the stored-program computers. It became attention when they relieved programmers of tasks such as memorizing numeric codes. However, by the late 1950s, their use replaces by higher-level languages in the search for greater programming productivity. ASM is still used today for direct hardware manipulation, gaining access to specialized processor instructions, and dealing with critical performance issues. Typical applications of device drivers, low-level embedded systems, and real-time systems include their use in various contexts. Many commercial applications, such as large amounts of IBM mainframe software, were also written in assembly languages. Microcomputers relied heavily on hand-coded assembly languages in their early stages. It was due to a lack of high-level language compilers designed to operate on microcomputers. Another reason for the assembly language set at the time was that they had many advantages, such as small size, high speed, low overhead, and high reliability.

### How Assembly Language Works?

Assembly language uses a series of mnemonic codes to represent machine language instructions. A programmer writes these codes in a text editor, and then an assembler program assembles them into machine language instructions that the computer can understand and execute.

The process of assembling the code involves converting the mnemonic codes into their binary equivalents, which the CPU can execute directly. The assembler program stores the resulting machine language instructions in an object or executable file that the CPU can load into memory and execute.

It provides programmers with direct hardware control, enabling them to access the low-level components of a computer system, such as the CPU, memory, and input/output devices. This makes it a valuable language for low-level programming tasks such as writing device drivers and real-time and embedded systems.

Additionally, assembler language is highly optimized for performance, allowing programmers to fine-tune the code to take full advantage of the underlying hardware. Assembly language programs can be minimal in size and require fewer resources than higher-level language programs, making them ideal for memory-constrained environments.

However, writing code in assembly language can be time-consuming and requires a deep understanding of hardware architecture. Therefore, Programmers often use assembly language only for specific tasks where its unique strengths are needed, while they use higher-level languages for other programming tasks.

### Assemblers

It is translated into machine language by assemblers. An assembler allows software and application developers to access, operate, and manage a hardware architecture and components of a computer. It is also known as an assembly language compiler.

### Types of Assemblers

There are two main types of assemblers used in computer programming:

1. **Single-pass assembler**: A single-pass assembler, also known as a one-pass translation, performs a complete scan of the source program input to the assembler or its equivalent representation and translates each statement based on the declaration. In other words, a single-pass assembler reads the entire source code and generates the corresponding machine code in one pass without requiring multiple iterations. It isolates the label, mnemonics, and operand field of the system. It validates the code instructions by looking them up in a mnemonic code table. It enters the symbol observed in the label field and addresses the available text’s machine word in the symbol table. This pass is fast and effective, and no need to construct the intermediate code.
2. **Multi-pass assembler:** An assembler must go through assembly language several times to generate object code. This final pass is called a synthesis pass, and this assembler requires any form of an intermediate code to create each pass every time. It is comparatively slower than a single-pass assembler, but some actions can perform more than once means duplicated.

### Components of Assembly Language: Understanding the Building Blocks of Low-Level Programming

Let’s examine a few assembly language components and how they convert to machine code.

* **Command:**A command is an instruction in assembly code that tells the assembler what action to take. Assembly language commands frequently use abbreviations to keep the terminology short while also using self-descriptive abbreviations, such as “ADD” for addition and “MOV” for data transfer.
* **Label:**A label is a word, number, or symbol that the assembly code uses to locate the date or instruction to use.
* **Operand:**A variable or piece of data is an operand, which the assembler can manipulate. Users can tell the assembly language which data to apply whatever commands on that line by including operands in instructions.
* **Mnemonic:**A mnemonic is a name given to a machine function or an abbreviation for an assembly language operation. In assembly, each mnemonic represents a machine instruction. An example of one of these machine instructions add. Mul, lea, and CMP are some other examples.
* **Registers**: The processor can operate on numeric values (numbers) but must first save somewhere. The data is now kept in memory, inside the instruction opcode, or in special on-chip memory placed directly in the processor, known as registers. To work with a value, users don’t need to address it by address; instead, special mnemonic “names” such as ax on x86, A on Z80, or r0 on ARM are in use.
* **Macro**: An assembly language macro is a template high-heel format that presents a series or pattern of statements. This set of assembly language statements is by several programs. A macro facility uses to interpret macro definitions, and a macro call inserts into the source code where “normal” assembly code would have gone instead of the macro set of statements.
* **Directive**: A directive is a kind of instruction that informs the assembler how to change a setting or perform an action. An assembly language’s syntax frequently requires a specific character, such as a period, at the start of a directive. It instructs the assembler that the following text is a directive to follow.

### Assembly Language and High-Level Language

|  |  |
| --- | --- |
| **Assembly Language** | **High-Level language** |
| Assembly language is machine-dependent. It is a low-level computer language that allows users to create a wide range of programs by representing a set of instructions with alphanumeric codes rather than numeric codes.  Example: ARM, MIPS, x86, Z80 | A high-level language is machine-independent. A high-level language is user-friendly, uses variables and functions, and is not dependent on the computer’s architecture.    Example: C, C++, Python, Javascript. |
| The assembly language requires an assembler to convert. | A high-level language conversion requires the use of an interpreter or compiler. |
| It is closely related to hardware and is thus used to write hardware programs. | It has no hardware interactions and is only used to write software application programs. |
| Faster program execution. | Program execution is slow. |

### Debugging Assembly language code

When debugging assembly language programs, effective debugging techniques include:

**1. Debugger:**A debugger is a part of the software that allows you to step through your assembly language program, examine the contents of registers and memory, and set breakpoints to stop the program at specific points for further examination.

**2. Instruction set for assembly language:**Understanding the assembly language instruction set, including the operation of each instruction, the registers it uses, and the flags it sets, will aid in understanding the program flow and the cause of the problem.

**3. Print statements or data output to a file:**It can be applied to inspect the values of variables and registers at specific points in the program to determine how the program behaves and where the problem is.

**4. Examining memory:**It is possible to determine whether the problem arises from incorrect memory access or incorrectly initialized data by inspecting the contents of the memory.

**5. Using a simulator:**The assembly code operates in a simulated environment without the requirement for the target hardware using a simulator.

**6. Profiling Tools:**Profiling tools can detect performance bottlenecks, memory leaks, and hotspots in assembly code.

### Application

Real-world applications are one of the most valuable applications of Assembler language programming.

* **LED (LIGHT EMITTING DIODE):**Light Emitting Diodes (LED) are the most common components, typically used to display the digital states of pins. Typical uses of LEDs include alarm devices, timers, and confirmation of user input, such as a mouse click or keystroke.
* **8051 Trainer Kit:**The PS-TRAINER-8051 trainer kit recommends for learning and developing MCU designs from Intel and NXP. It can connect to a PC’s 101/104 keyboard to enter user programs written in Assembly languages.
* **Interfacing LED with 8051:**The LED in the 8051 Trainer Board works by turning on an LED, then off, and then looping back to START. However, because the microcontroller’s operating speed is very high, the flashing frequency will be sufficiently fast to be detected by the human eye.

### Assembly Program to ON and OFF LED using 8051

|  |  |  |
| --- | --- | --- |
| **MEMORY ADDRESS** | **OPCODE** | **MNEMONICS** |
| 9100 | 74 55 | L1:MOV A,#FF |
| 9102 | 7C | MOV B0,A |
| 9104 | 12 91 0F | LCALL DELAY |
| 9107 | 74 00 | MOV A,#00 |
| 9109 | 7C | MOV B0,A |
| 910b | 12 91 0F | LCALL DELAY |
| 910D | 80 F9 | SJMP L1 |
| 910F | 7D 05 | DELAY: MOV R5,#05 |
| 9111 | 7C FF | H3 MOV R4,#FF |
| 9113 | 7B FF | H2 MOV R3,#FF |
| 9115 | DB FE | H1: DJNZ R3,H1 |
| 9117 | DC FA | DJNZ R4,H2 |
| 9119 | DD F6 | DJNZ R5,H3 |
| 911B | 22 | RET |

### Future of Assembly language

1. **Specialized applications:** It will continue to be important in technical applications such as embedded systems, real-time systems, and device drivers that require low-level hardware control and high performance.
2. **Limited use:** Assembly language is likely to become more specialized and limited to specific areas where its unique strengths are required.
3. **Higher-level languages:** As higher-level languages evolve and become more efficient, the need for assembler language programming in many applications will likely reduce.
4. **Compiler optimization:** Advances in compilers and optimization techniques have made it possible to write high-performance code in higher-level languages, reducing the need for assembler language programming.
5. **Interpreted languages:** The increasing popularity of interpreted and just-in-time compiled languages, such as Python and JavaScript, has further reduced the need for low-level programming.
6. **Computer science education:** Assembler language will remain essential to computer science and engineering education, providing a deeper understanding of computer architecture and low-level programming.

While the use of symbolic machine code may decline in some areas, it will continue to play a critical role in specialized applications where its strengths in performance and hardware control are required.

### Importance of assembly language

It helps programmers in coding human-readable code that is relatively identical to machine language. Machine language is difficult to understand and read as it is just a series of numbers. Assembler language gives complete control over what tasks a computer performs.

ASM language is an official language that connects hardware and software. The following are the main characteristics of ASM languages that make users suitable for modern programming:

1. **Direct hardware control**: Assembler language allows programmers to have direct control over the hardware of a computer system, such as the CPU, memory, and input/output devices. This makes it a valuable language for low-level programming tasks, such as writing device drivers.
2. **High performance**: Programs written in ASM language can be highly optimized for performance, making them ideal for applications that require fast execution times, such as real-time systems and scientific simulations.
3. **Small code size**: Assembly language programs tend to be smaller than programs written in higher-level languages, which can be advantageous in memory-constrained environments, such as embedded systems.
4. **Portability**: Programmers can write Assembler language code to be portable across different computer architectures, enabling them to adapt programs to run on other hardware platforms efficiently.
5. **Education**: Symbolic machine code is often used in computer science and engineering courses to teach students about computer architecture and system-level programming.
6. **Human readable:** Assembler languages use words, numbers, and other textual components to create instructions to understand, remember, and recreate. As a result, symbolic machine code is relatively simple to use and learn while still forming an executable command for the machine.
7. **Assist debugging:** The language is useful for error handling because programmers can read every line to understand how the code fails. It is more potent than the debugger in typical debugging because assembler language can be used to find issues with pointer arithmetic and memory corruption.

**Example:**

Find the below steps to print “Hello, World” on Windows

1. Open the notepad.
2. Write below code

global \_main

extern \_printf

section .text

\_main:

push     message

call      \_printf

add      esp, 4

ret

message:

db ‘Hello, World!’, 10, 0

1. Save the file with any name, for example, XYZ.asm, with the extension.’ asm.’
2. The above file must compile using the NASM assembler (Netwide Assembler).
3. Run the command nasm –f win32 XYZ.asm.
4. Nasm then generates one object file, XYZ.obj, which contains machine code but not executable code.
5. Minimal GNU, which includes the GCC compiler, is used to create the executable file for Windows.
6. Run the command gcc –o XYZ.exe XYZ.obj
7. Execute the executable file now, ‘XYZ.’
8. It will print “Hello, World!” as the output.

### Reason why we should learn assembly language

The learning of ASM is still helpful for programmers. It helps in taking complete control over the system and its resources.

* By learning ASM language, the programmer can write the code to access registers and retrieve the memory address of pointers and values. It mainly helps in speed optimization, which increases efficiency and performance.
* ASM language learning helps in understanding the processor and memory functions. If a programmer is writing a program that requires a compiler, the programmer requires a complete understanding of the processor.
* ASM language helps in understanding the work of processors and memory. It is a cryptic and symbolic language.
* ASM Language helps in contacting the hardware directly. This language forms mainly on computer architecture, and it recognizes a specific type of processor and is different for different CPUs.
* ASM language refers to transparency compared to other high-level languages. It has a small number of operations, but it helps understand the algorithms and other control flows. It makes the code less complex and easy to debug as well.

### Features of Assembly Language

The following are the features of the ASM:

1. It can use mnemonic than numeric operation code, and it further reports any error in the code.
2. This language helps specify the symbolic operand, which means it does not need to specify the machine address of that operand. It represents in the form of a symbol.
3. The data declare by using decimal notation.
4. Errors and bugs are easily identified and resolved.
5. ASM language is simple to understand and execute.

### Tips and Tricks

It could be supplementary reading for assembler language Programming.

1. Using instructions with very few bytes.
2. Using memory variables to implement.
3. According to a basic rule in ASM language programming. The register operation is much faster than the memory operation.
4. An x86 processor uses little-endian order to store and retrieve data from memory (low to high). The least significant byte saves at the data’s first memory address. The Remaining bytes are in the memory positions that follow.
5. Use a macro to generate memory blocks during assembly rather than at runtime.
6. All loop logic encapsulates within the loop body.
7. Regarding the procedure, develop a simple calling interface.
8. Don’t use the OFFSET operator to get the address of an indirect operand pointing to an object because OFFSET can only take the address of a variable defined in the data segment.
9. In ASM language programming, an integer variable can be either signed (SBYTE, SWORD, and SDWORD) or unsigned (BYTE, WORD, and DWORD).
10. Ignore using multiple RET to build procedures because ASMlanguage programming directly defines memory addresses and any labels. Hence, a direct jump without using CALL or INVOKE would be ideal.

### ****Advantages and Disadvantages****

Mentioned are some advantages and disadvantages:

|  |  |
| --- | --- |
| **Advantages** | **Disadvantages** |
| It allows complex jobs to execute in a standard way. | Require a lot of time and endeavor  to write the code for the same. |
| It is memory efficient, as it requires less memory | It needs more size or memory of the computer to run more programs written in ASM Language |
| It is faster, as its execution time is less. | The syntax is difficult to remember. |
| It is mainly hardware-oriented. | It has a lack of portability of programs between [**different computer architectures**](https://www.educba.com/types-of-computer-architecture/). |
| It requires less instruction to get the result. | It is very complex and tough to understand. |
| It utilizes for critical jobs. | It is not suitable for a few industries |
| It is not necessary to keep track of memory locations. | It is a challenging task to identify and rectify the errors in the code. |
| It is a low-level embedded system. | It is not portable. |

### Final Thoughts

Assembly language is essential for understanding computer architecture and programs for programmers. The programmers mainly used many other programming languages for application development and software, but ASM language is also significant. It helps programmers to achieve a lot if they implement the ASM language. Assemblies contain a lot of metadata, that as version numbers, localization details, and other product details. It is an important part and displays to the user after being digitally signed.

If an individual wants to know how the system works and the processor as well, then ASM language is the one that solves the purpose. It helps in all aspects, from understanding the program’s algorithm to the processor working and registering the computer’s register. It depends on individual choice with which language to continue.

#### Why is it called ASM Code?

**Answer:** “Assembly language” means a kind of low-level programming language that directly controls a computer’s hardware. The term “ASM language” refers to the process of translating mnemonics (abbreviations or acronyms) into machine code instructions (also known as “assembler”). In comparison, higher-level languages like C and Python translate into machine code by a compiler or interpreter.

#### ****What is the basic syntax of****ASM Code****?****

**Answer:** Any ASM language program consists of three sections

**Data section:**The data section utilizes to declare initialized data or constants. This information does not change during execution. In this section, you can confirm various constant values, file names, buffer sizes, etc.

The syntax for initiating the data section

section. Data

**BSS section:**The bss section is where variables are declared.

The syntax for the BSS section

section.bss

**Text Section:**The text section is where the source code is secured. ASM language instructions and assembler directives may include in the text section.

The syntax for initiating the Text section

section.text

global \_start

\_start:

The global\_start declaration shows the kernel where program execution begins.

**Syntax of Assembly Language Statements**

[label] mnemonic [operands] [comment]

ASM language statements enter one per line. A basic instruction consists of two parts: the name of the commands (or mnemonic) to execute and the operands or command parameters.

**DIFFERENCE BETWEEN C LANGUAGE AND ASSEMBLY LANGUAGE**

|  |  |
| --- | --- |
| **C language** | **Assembly language** |
| The C programming language is portable and does not require a specific platform. The code written in C can easily reuse on another platform. | Assembly does not provide portability or processor-specific source code because assembly instructions depend on the processor architecture. |
| C is convenient to use for making more complex programs. | ASM language is much more flexible, allowing you to work with memory, interrupts, micro-registers, etc. |

**Q4. What is another name for an ASM language?**

**Answer:** ASM language (assembler or symbolic machine code), commonly abbreviated as ASM or asm, is any low-level programming language with a solid correspondence between the instructions in the language and the architecture’s machine code instructions.

**What are the registers in the ASM language?**

 The processor includes a few internal memory storage locations known as registers to speed up processor operations. It stores data elements for processing without requiring memory access. The number on the processor chip is limited.

**Processor Registers:** A processor register (CPU register) is one of a small group of data storage locations built into a computer processor.

The registers classify into three types.

**General registers:** General Registers, also known as General Purpose Registers, that store data and addresses.

The general registers are categorized as follows:

**DX (Data register):**The AX register in the 8086 microprocessor is a 16-bit data register split into two 8-bit registers, AH and AL. It is commonly used in 8-bit instructions, can be utilized as a port number in I/O operations, and is employed for multiplication and division with specific instructions for each operation.

**Pointer and Index register:** These registers store data and instruction offsets. The offset is the distance between a variable, label, or instruction and its base segment.

**Control Registers:**Intel-architecture processors have a collection of control registers used to configure the processor at run time (such as switching between execution modes).

**Segment Registers:** Segments are programmatically defined areas that contain data, code, and stack.

**The resources for Learning ASM code?**

ARM, MIPS, and X86 are widely used assembler language programs.

**Courses:**

**1. Learn ASM Language by Making Games for the Atari 2600**

This ASM language course online will teach you the fundamentals of programming games using 6502 ASM language.

**2. ASM Programming**

This course teaches you how to create applications using ASM language programming and learn the fundamentals of project creation, the structure of assembly code, and advanced topics such as function recursion and system calls.

**Tools**

1. **Assembler:**The Assembler program helps to generate ASMlanguage mnemonics for instructions into binary codes.
2. **Linker:**A linker is a program that joins multiple object files into a single large object file.
3. **Locator:**A locator is a program used for a particular address that determines the sections of object code to load into memory.

**How is For Loop implemented in assembly language to calculate the sum?**

**Answer:** The “LOOP target” statement performs two functions:

* decrement ECX
* if ECX != 0, jump to a target

To evaluate sum of n+(n-1)+…+2+1

mov ecx, n

xor eax, eax

L1:

add eax, ecx

loop L1

mov sum, eax

**What is the primary function of an assembler?**

Assembler is a program that converts ASM language to machine language. An assembler’s primary function is to assign addresses to labels. It converts the basic commands and operations from assembly code into binary code that can recognize by a specific type of processor. Assemblers, like compilers, create executable code.