Fundamentals of Computer Technology- 1st Course

اساسيات تكنولوجيا الحاسوب

First Grade

Software Branch

Dr. Inaam S. Naser

Overview

Today's world is an information-rich world and it has become a necessity for everyone to know about computers. A computer is an electronic data processing device, which accepts and stores data input, processes the data input, and generates the output in a required format.

Functionalities of a Computer

If we look at it in a very broad sense, any digital computer carries out the following five functions:

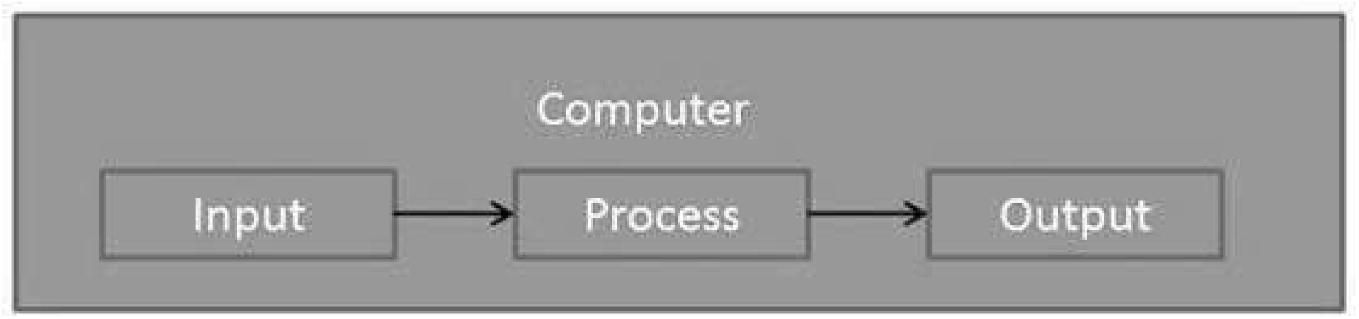
Step 1 - Takes data as input.

Step 2 - Stores the data/instructions in its memory and uses them as required.

Step 3 - Processes the data and converts it into useful information.

Step 4 - Generates the output.

Step 5 - Controls all the above four steps.



Advantages of Computers

Following are certain advantages of computers.

High Speed

- ✓ Computer is a very fast device.
- ✓ It is capable of performing calculation of very large amount of data.
- ✓ The computer has units of speed in microsecond, nanosecond, and even the picosecond.
- ✓ It can perform millions of calculations in a few seconds as compared to man who will spend many months to perform the same task.

Accuracy

- ✓ In addition to being very fast, computers are very accurate.
- ✓ The calculations are 100% error free.
- ✓ Computers perform all jobs with 100% accuracy provided that the input is correct.

Storage Capability

- ✓ Memory is a very important characteristic of computers.
- ✓ A computer has much more storage capacity than human beings.
- ✓ It can store large amount of data.
- ✓ It can store any type of data such as images, videos, text, audio, etc.

Diligence

- ✓ Unlike human beings, a computer is free from monotony, tiredness, and lack of concentration.
- ✓ It can work continuously without any error and boredom.
- ✓ It can perform repeated tasks with the same speed and accuracy.

Versatility

- ✓ A computer is a very versatile machine.
- ✓ A computer is very flexible in performing the jobs to be done.
- ✓ This machine can be used to solve the problems related to various fields.
- ✓ At one instance, it may be solving a complex scientific problem and the very next moment it may be playing a card game.

Reliability

- ✓ A computer is a reliable machine.
- ✓ Modern electronic components have long lives.
- ✓ Computers are designed to make maintenance easy.

Automation

- ✓ Computer is an automatic machine.
- ✓ Automation is the ability to perform a given task automatically. Once the computer receives a program i.e., the program is stored in the computer

memory, then the program and instruction can control the program execution without human interaction.

Reduction in Paper Work and Cost

- ✓ The use of computers for data processing in an organization leads to reduction in paper work and results in speeding up the process.
- ✓ As data in electronic files can be retrieved as and when required, the problem of maintenance of large number of paper files gets reduced.
- ✓ Though the initial investment for installing a computer is high, it substantially reduces the cost of each of its transaction. Modern electronic components have long lives.

Disadvantages of Computers

Following are certain disadvantages of computers.

No I.Q.

- ✓ A computer is a machine that has no intelligence to perform any task.
- ✓ Each instruction has to be given to the computer.
- ✓ A computer cannot take any decision on its own.

Dependency

- ✓ It functions as per the user's instruction, thus it is fully dependent on humans.

Environment

- ✓ The operating environment of the computer should be dust free and suitable.

No Feeling

- ✓ Computers have no feelings or emotions.
- ✓ It cannot make judgment based on feeling, taste, experience, and knowledge unlike humans.

Software

Software, simply are the computer programs. The instructions given to the computer in the form of a program is called Software. Software is the set of programs, which are used for different purposes. All the programs used in computer to perform specific task is called Software.

Software is defined as a collection of computer programs, procedures, rules, and data.

Software types

Based on the goal, computer software can be divided into:

1. **Application software** uses the computer system to perform special functions beyond the basic operation of the computer itself. There are many different types of application software because the range of tasks that can be performed with a modern computer is so large. Ex. (Microsoft office, Photoshop, etc.)
2. **System software** manages hardware behavior, as to provide basic functionalities that are required by users, or for other software to run properly, if at all. System software is also designed for providing a platform for running application software, and it includes the following:

   - ***Operating systems*** are essential collections of software that manage resources and provide common services for other software that runs "on top" of them. Ex. (DOS, Windows XP, Windows Vista, Unix/Linux, MAC/OS X, Windows7 etc.)

- ***Device drivers*** operate or control a particular type of device that is attached to a computer. Each device needs at least one corresponding device driver; because a computer typically has at minimum at least one input device and at least one output device, a computer typically needs more than one device driver.

- ***Utilities*** are computer programs designed to assist users in the maintenance and care of their computers. Ex. (Windows Explorer (File/Folder Management), Windows Media Player, Anti-Virus Utilities, Disk Defragmentation, Disk Clean, BackUp, WinZip, WinRAR etc.)

3. **Malicious software**, or malware, is software that is developed to harm or disrupt computers. Malware is closely associated with computer-related crimes, though some malicious programs may have been designed as practical jokes.

Software Characteristics

Software characteristics are classified into six major components: (according to the ISO/IEC 9126 standards)

- *ISO/IEC 9126 standards (an international standard for the evaluation of software) is an international standard for the evaluation of software quality. It has been replaced by ISO/IEC 25010:2011.*

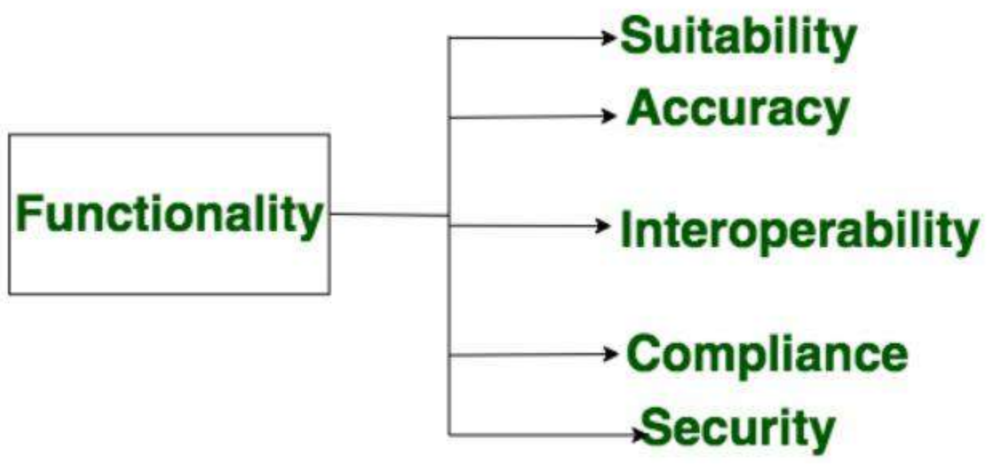
1. Functionality

It refers to the degree of performance of the software against its intended purpose.

Examples of functionality in software include:

- Data storage and retrieval
- Data processing and manipulation
- User interface and navigation
- Communication and networking
- Security and access control
- Reporting and visualization
- Automation and scripting

Required functions are:



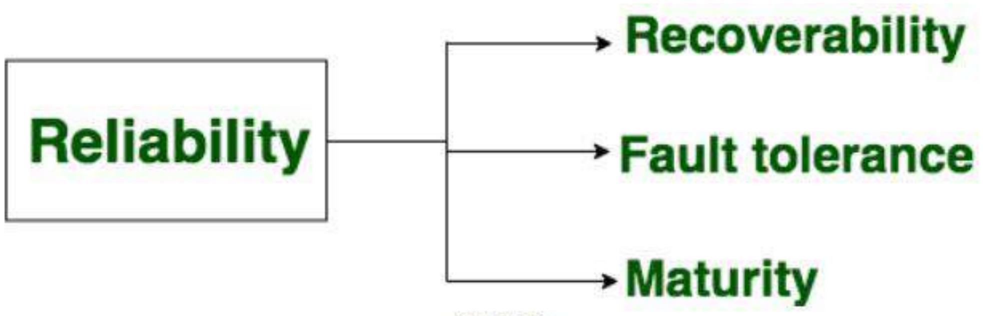
2. Reliability

A set of attributes that bears on the capability of software to maintain its level of performance under the given condition for a stated period of time.

Examples of factors that can affect the reliability of software include:

1. Bugs and errors in the code
2. Lack of testing and validation
3. Poorly designed algorithms and data structures
4. Inadequate error handling and recovery
5. Incompatibilities with other software or hardware

Required functions are:



Reliability

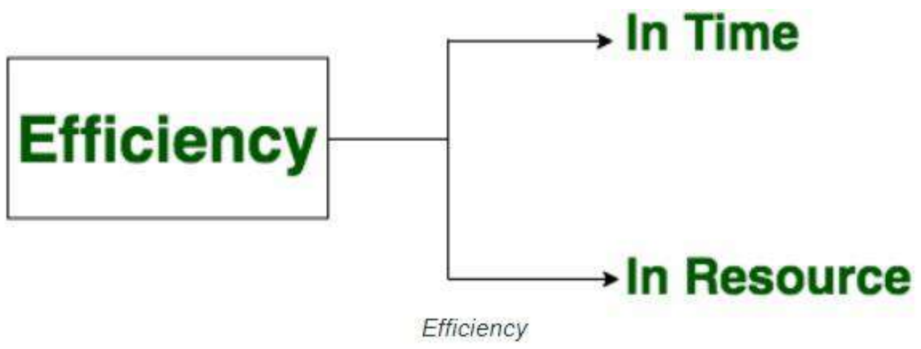
3. Efficiency

It refers to the ability of the software to use system resources in the most effective and efficient manner. The software should make effective use of storage space and executive command as per desired timing requirements.

Examples of factors that can affect the efficiency of the software include:

1. Poorly designed algorithms and data structures
2. Inefficient use of memory and processing power
3. High network latency or bandwidth usage
4. Unnecessary processing or computation
5. Unoptimized code

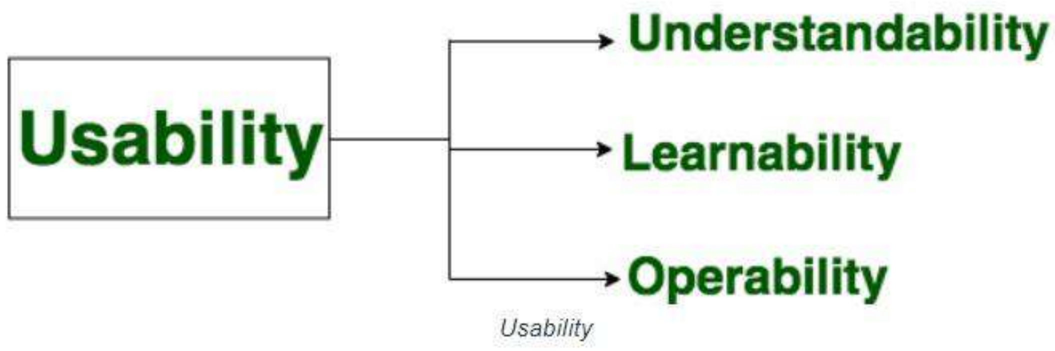
Required functions are:



4. **Usability**

It refers to the extent to which the software can be used with ease. The amount of effort or time required to learn how to use the software.

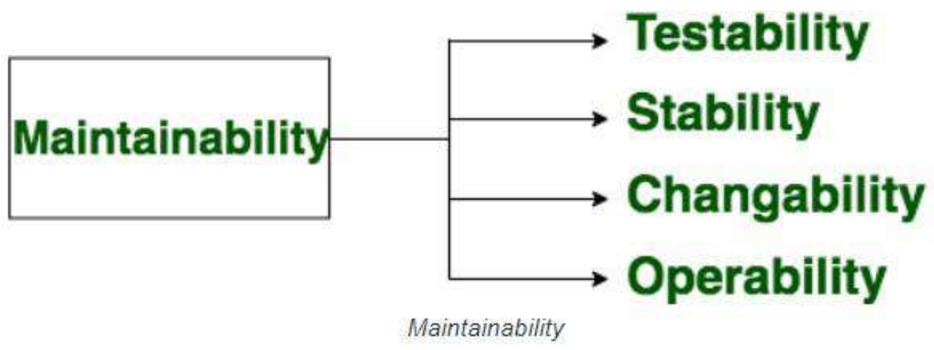
Required functions are:



5. **Maintainability**

It refers to the ease with which modifications can be made in a software system to extend its functionality, improve its performance, or correct errors.

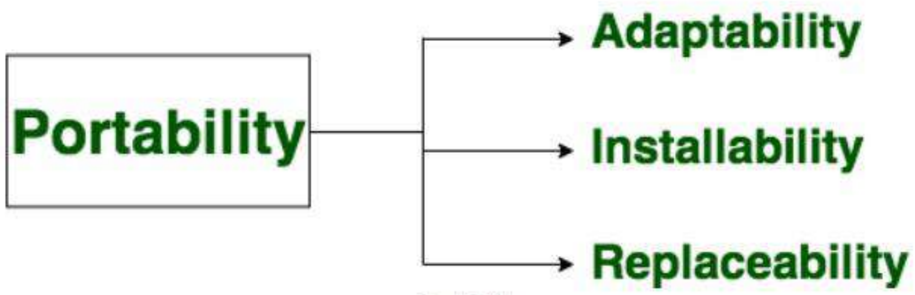
Required functions are:



6. **Portability**

A set of attributes that bears on the ability of software to be transferred from one environment to another, without minimum changes.

Required functions are:



Portability

Computer Languages Basics:

a) Low Level Language

i) Machine language:

- ✓ These language instructions are directly executed by CPU
- ✓ Machine language consists of binary and hexadecimal characters.

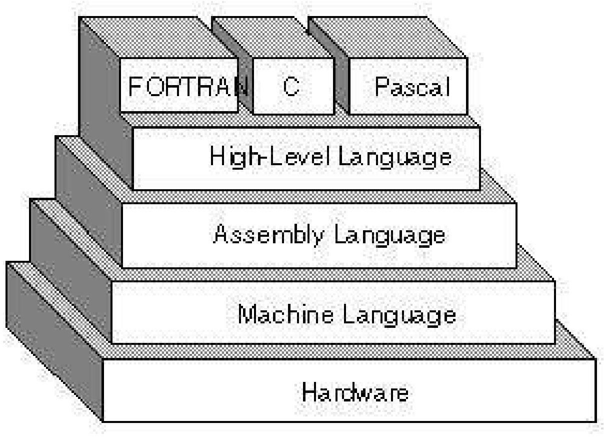
ii) Assembly language:

- ✓ An assembly language is a type of low-level programming language that is intended to communicate directly with a computer’s hardware.
- ✓ Assembly languages are designed to be readable by humans.

B) High Level Language: The user friendly language, more natural language than assembly language.

Examples:

COBOL (COmmon Business Oriented Language), FORTRAN (FORmula TRANslation), BASIC (Beginner's All-purpose Symbolic Instruction Code), C, C++, Python, JavaScript etc.



Assembler is needed to convert assembly language into machine language

Complier is needed to convert high level to machine language

Understanding Applications

You may have heard people talking about using a program, an application, or an app. But what exactly does that mean? Simply put, an app is a type of software that allows you to perform specific tasks. Applications for desktop or laptop computers are sometimes called desktop applications, while those for mobile devices are called mobile apps.

When you open an application, it runs inside the operating system until you close it. Most of the time, you will have more than one application open at the same time, which is known as multi-tasking.

App is a common term for an application, especially for simple applications that can be downloaded inexpensively or even for free. Many apps are also available for mobile devices and even some TVs.

Desktop applications

There are countless desktop applications, and they fall into several categories. Some are more full featured (like Microsoft Word), while others may only do one or two things (like a clock or calendar app). Below are just a few types of applications you might use.

- ***Word processors:*** A word processor allows you to write a letter, design a flyer, and create many other types of documents. The most well-known word processor is Microsoft Word.

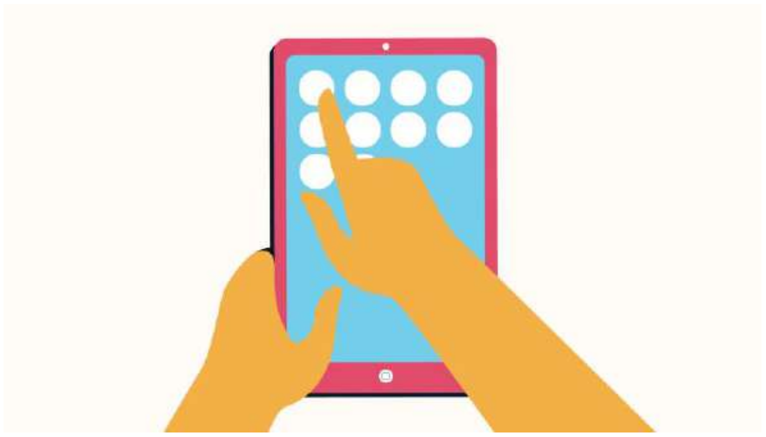
- ***Web browsers:*** A web browser is the tool you use to access the Internet. Most computers come with a web browser pre-installed, but you can also download a different one if you prefer. Examples of browsers include Internet Explorer, Mozilla Firefox, Google Chrome, and Safari.

- ***Media players:*** If you want to listen to MP3s or watch movies you've downloaded, you'll need to use a media player. Windows Media Player and iTunes are popular media players.

- ***Games:*** There are many types of games you can play on your computer. They range from card games like Solitaire to action games like Halo. Many action games require a lot of computing power, so they may not work unless you have a newer computer.

Mobile applications

Desktop and laptop computers aren't the only devices that can run applications. You can also download apps for mobile devices like smartphones and tablets. Here are a few examples of mobile apps.



Gmail: You can use the Gmail app to easily view and send emails from your mobile device. It's available for Android and iOS devices.

Instagram: You can use Instagram to quickly share photos with your friends and family. It's available for Android and iOS.

Duolingo: With a combination of quizzes, games, and other activities, this app can help you learn new languages. It's available for Android and iOS.

Understanding the Cloud

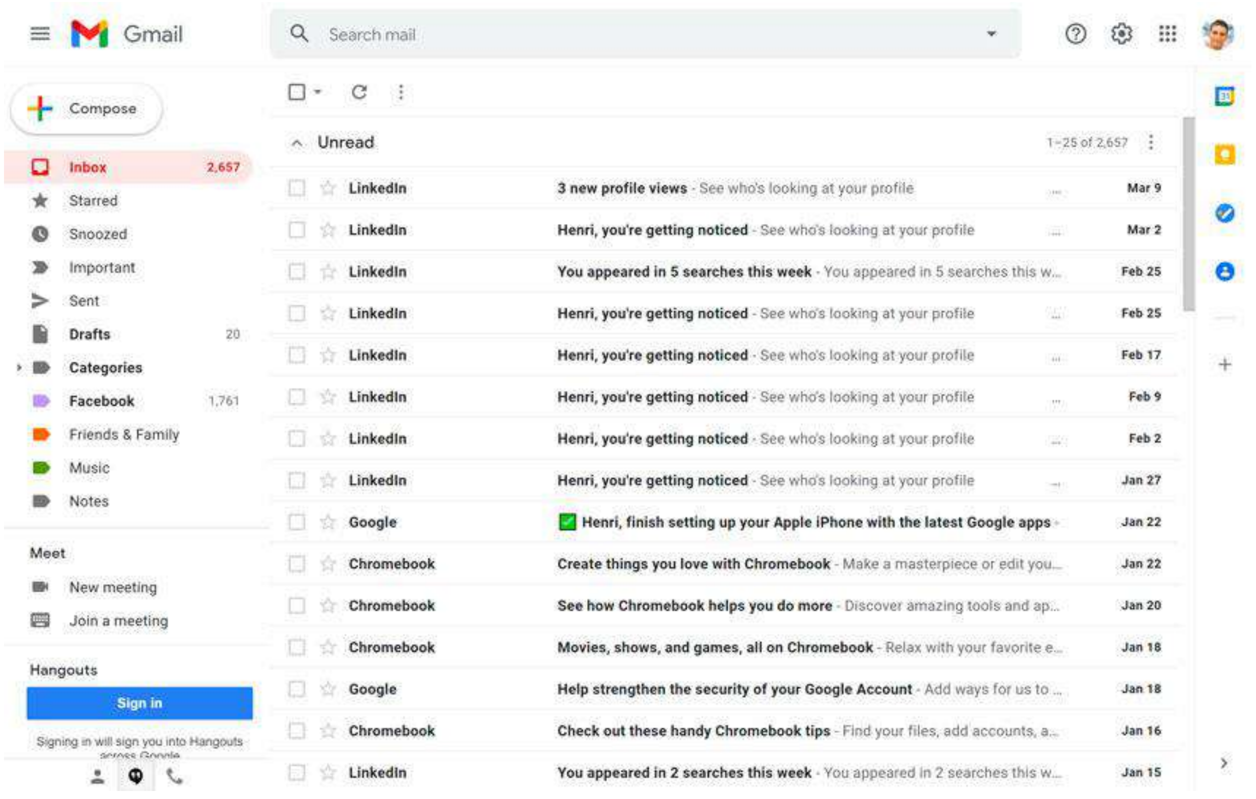
What is the cloud?

You may have heard people using terms like the cloud, cloud computing, or cloud storage. But what exactly is the cloud?

Simply put, the cloud is the Internet—more specifically, it's all of the things you can access remotely over the Internet. When something is in the cloud, it means it's stored on Internet servers instead of your computer's hard drive.

Why use the cloud?

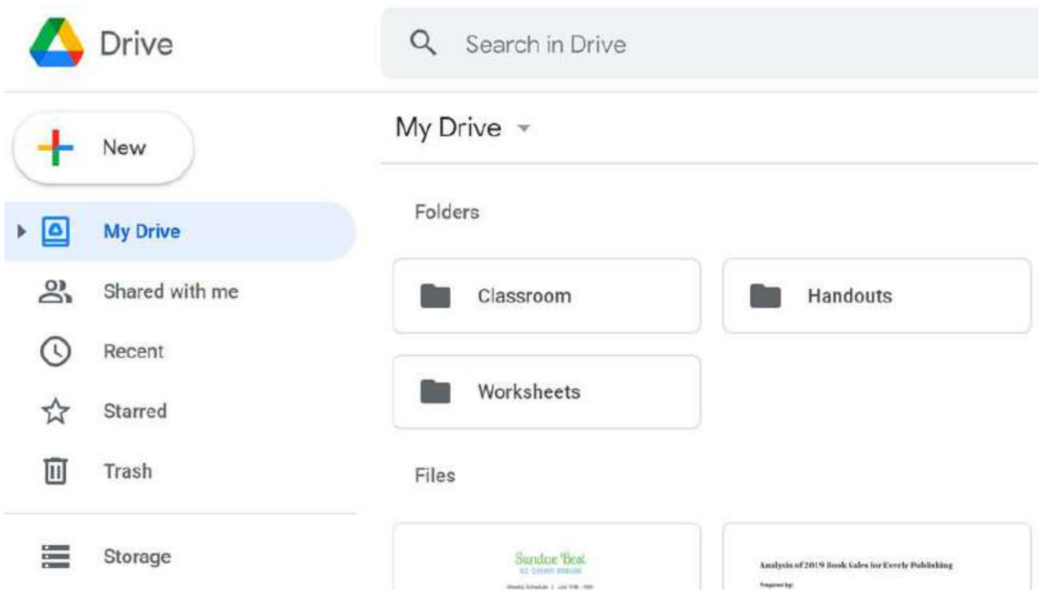
Some of the main reasons to use the cloud are convenience and reliability. For example, if you've ever used a web-based email service, such as Gmail or Yahoo! Mail, you've already used the cloud. All of the emails in a web-based service are stored on servers rather than on your computer's hard drive. This means you can access your email from any computer with an Internet connection. It also means you'll be able to recover your emails if something happens to your computer.



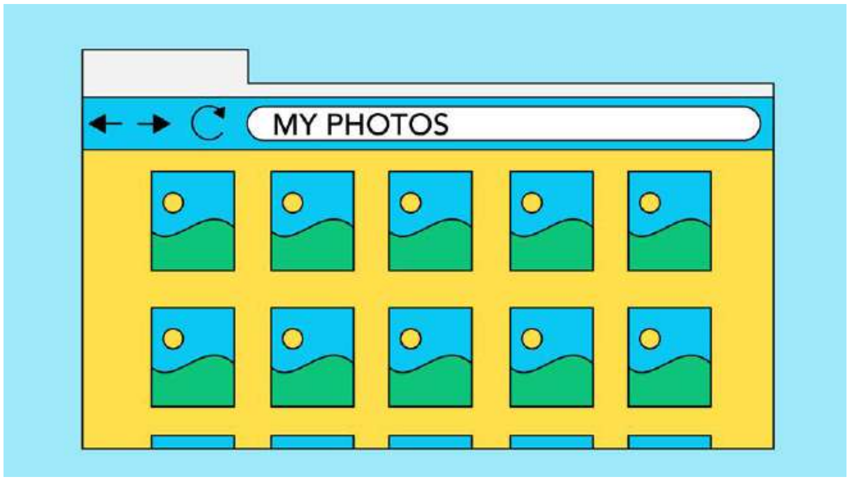
LinkedIn	**3 new profile views** - See who's looking at your profile	Mar 9
LinkedIn	**Henri, you're getting noticed** - See who's looking at your profile	Mar 2
LinkedIn	**You appeared in 5 searches this week** - You appeared in 5 searches this w...	Feb 25
LinkedIn	**Henri, you're getting noticed** - See who's looking at your profile	Feb 25
LinkedIn	**Henri, you're getting noticed** - See who's looking at your profile	Feb 17
LinkedIn	**Henri, you're getting noticed** - See who's looking at your profile	Feb 9
LinkedIn	**Henri, you're getting noticed** - See who's looking at your profile	Feb 2
LinkedIn	**Henri, you're getting noticed** - See who's looking at your profile	Jan 27
Google	✅ **Henri, finish setting up your Apple iPhone with the latest Google apps** -	Jan 22
Chromebook	**Create things you love with Chromebook** - Make a masterpiece or edit you...	Jan 22
Chromebook	**See how Chromebook helps you do more** - Discover amazing tools and ap...	Jan 20
Chromebook	**Movies, shows, and games, all on Chromebook** - Relax with your favorite e...	Jan 18
Google	**Help strengthen the security of your Google Account** - Add ways for us to ...	Jan 18
Chromebook	**Check out these handy Chromebook tips** - Find your files, add accounts, a...	Jan 16
LinkedIn	**You appeared in 2 searches this week** - You appeared in 2 searches this w...	Jan 15

Let's look at some of the most common reasons to use the cloud.

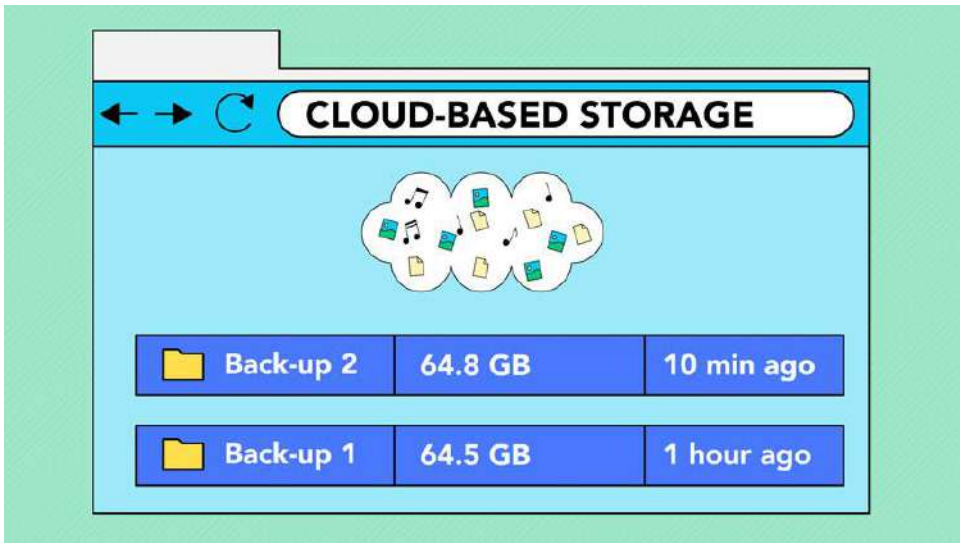
File storage: You can store all types of information in the cloud, including files and email. This means you can access these things from any computer or mobile device with an Internet connection, not just your home computer. Dropbox and Google Drive are some of the most popular cloud-based storage services.



File sharing: The cloud makes it easy to share files with several people at the same time. For example, you could upload several photos to a cloud-based photo service like Flickr or iCloud Photos, then quickly share them with friends and family.



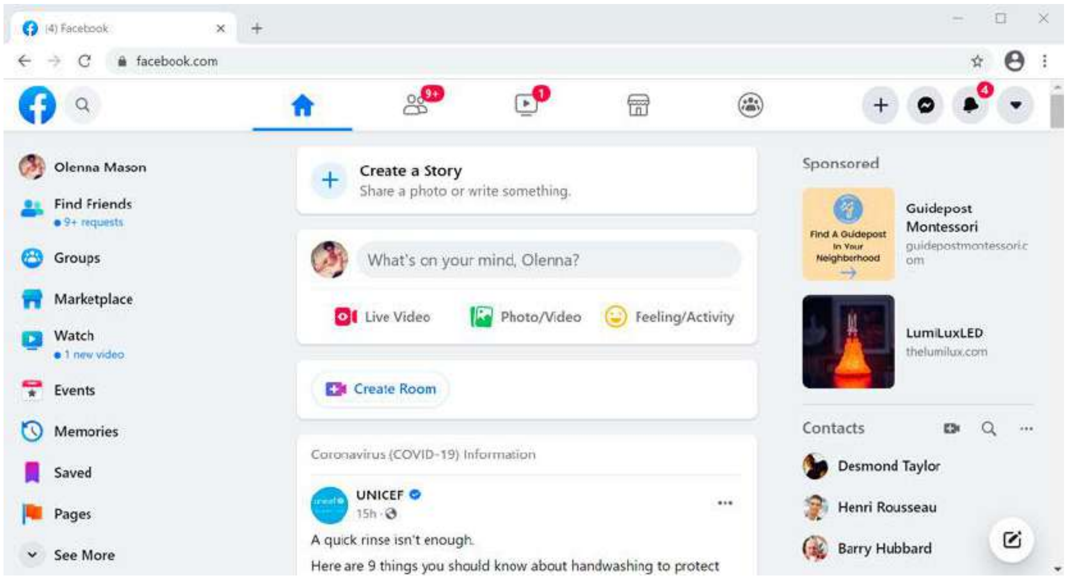
Backing up data: You can also use the cloud to protect your files. There are apps such as Carbonite that automatically back up your data to the cloud. This way, if your computer ever is lost, stolen, or damaged, you'll still be able to recover these files from the cloud.



What is a web app?

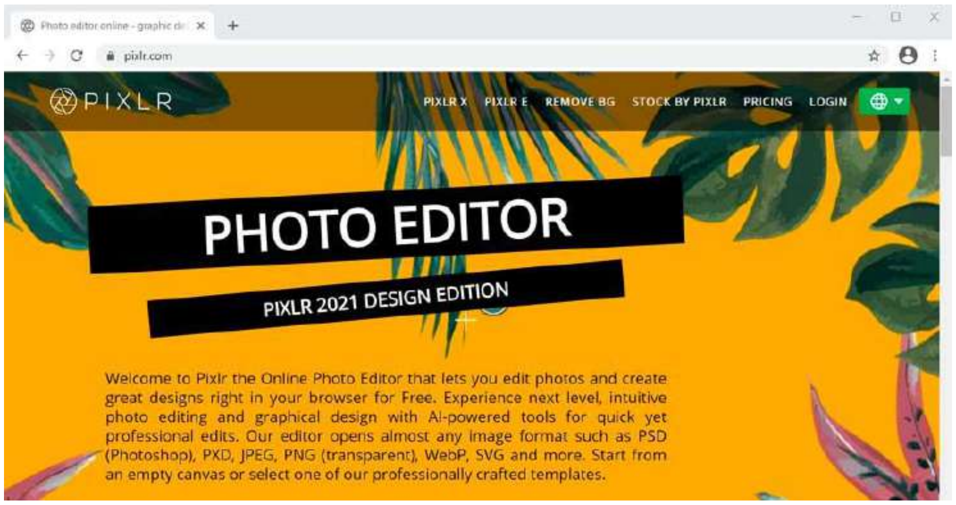
Previously, we talked about how desktop applications allow you to perform tasks on your computer. But there are also web applications—or web apps—that run in the cloud and do not need to be installed on your computer. Many of the most popular sites on the Internet are actually web apps. You may have even used a web app without realizing it! Let's take a look at some popular web apps.

Facebook: Facebook lets you create an online profile and interact with your friends. Profiles and conversations can be updated at any time, so Facebook uses web app technologies to keep the information up to date.



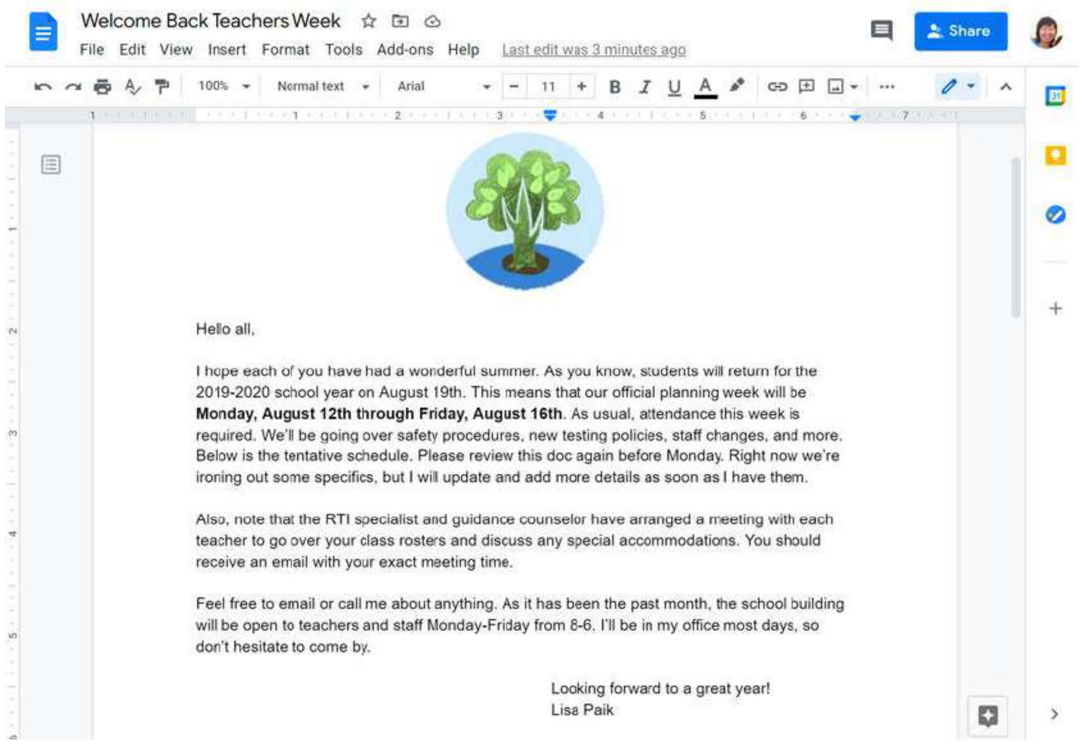
A quick rinse isn't enough.
Here are 9 things you should know about handwashing to protect

Pixlr: Pixlr is an image editing application that runs in your web browser. Much like Adobe Photoshop, it includes many advanced features, like color correction and sharpening tools.



Welcome to Pixlr the Online Photo Editor that lets you edit photos and create great designs right in your browser for Free. Experience next level, intuitive photo editing and graphical design with AI-powered tools for quick yet professional edits. Our editor opens almost any image format such as PSD (Photoshop), PXD, JPEG, PNG (transparent), WebP, SVG and more. Start from an empty canvas or select one of our professionally crafted templates.

Google Docs: Google Docs is an office suite that runs in your browser. Much like Microsoft Office, you can use it to create documents, spreadsheets, presentations, and more. And because the files are stored in the cloud, it's easy to share them with others.


Welcome Back Teachers Week
File Edit View Insert Format Tools Add-ons Help
Last edit was 3 minutes ago
Share
100%
Normal text
Arial
11

Hello all,

I hope each of you have had a wonderful summer. As you know, students will return for the 2019-2020 school year on August 19th. This means that our official planning week will be **Monday, August 12th through Friday, August 16th**. As usual, attendance this week is required. We'll be going over safety procedures, new testing policies, staff changes, and more. Below is the tentative schedule. Please review this doc again before Monday. Right now we're ironing out some specifics, but I will update and add more details as soon as I have them.

Also, note that the RTI specialist and guidance counselor have arranged a meeting with each teacher to go over your class rosters and discuss any special accommodations. You should receive an email with your exact meeting time.

Feel free to email or call me about anything. As it has been the past month, the school building will be open to teachers and staff Monday-Friday from 8-6. I'll be in my office most days, so don't hesitate to come by.

Looking forward to a great year!
Lisa Paik

Software development process

Software development is the process of designing, creating, testing, and maintaining different software applications. It involves the application of various principles and techniques from computer science, engineering and mathematical analysis. Software development aims to create efficient, reliable, and easy-to-use software.

The process of software development typically begins with the requirements-gathering phase.

- In this phase, the software application requirements are gathered from various stakeholders.
- These requirements are then analyzed and used to create a software design.
- And the software design is then implemented in code, which is then tested to ensure that it meets the requirements. Once the code is verified, it is deployed to the production environment.

Jobs That Use Software Development

Many jobs that use software development skills include software developers, engineers, and system administrators. These professionals use their skills to develop and maintain software applications, and they also use their skills to troubleshoot and fix software issues.

System Software

System software is the software that helps the computer system to function and perform all its tasks. It includes the operating system, which manages the hardware and software resources of the system, as well as the various utility programs that help to maintain and optimize the system.

System software jobs typically involve working with these different components to ensure they function correctly and efficiently. This can include troubleshooting and resolving issues and developing new features and enhancements. System software jobs typically require a solid technical background and problem-solving and analytical skills.

Programming Software

Many programming software jobs are available, from entry-level positions to more advanced functions.

Entry-level programming software jobs may involve writing or working with existing code to create new applications.

More advanced programming software jobs may involve developing new software or working on existing software to improve its performance.

But generally, programming software requires a high level of technical expertise and a deep understanding of how the software works. There are also many programming languages, so choosing a language that you are comfortable with is essential.

Application Software

Application software jobs are some of the most in-demand positions in the tech industry. As the world becomes more reliant on technology, businesses are looking for candidates with the skills to develop and maintain the software that powers their operations.

Application software developers are responsible for designing, creating, testing and maintaining the software that meets users' needs. They work with various programming languages and tools and must be able to troubleshoot issues arising during development.

The demand for qualified application software developers is expected to grow in the coming years, making this an excellent career choice for those with the right skills and training.

Programmers or Coders

Programmers or coders are responsible for creating and maintaining software applications. They use a variety of programming languages to write code that instructs computers to perform specific tasks. Coders also test and debug programs to ensure they are free of errors. In addition to writing code, programmers often collaborate with other software development team members, such as designers and system administrators, to create a compelling and cohesive final product.

Software Engineers

Software engineers are responsible for developing and maintaining software applications, and they work with various programming languages and tools to create, test and deploy software solutions. Along with writing code, software engineers also need to be able to solve complex problems and troubleshoot issues. As the demand for new and innovative software solutions continues to grow, so makes the demand for skilled software engineers.

Software Developers

A software developer job involves designing, creating, testing, and maintaining software applications. They may work in various industries, including computer science, engineering, information technology and business.

Most software developers work in office settings, though some may telecommute. They typically work full-time and may work evenings or weekends to meet deadlines.

Education requirements for software developers vary by employer, but most positions require at least a bachelor's degree in computer science or a related field. Some jobs may require certification in specific software programs.

Skills required for software developers include strong analytical and problem-solving abilities and experience in one or more programming languages. They must be able to work independently and as part of a team.

Software Development Methodologies

There are a variety of software development methodologies that can be used to create software applications. The most popular methods include the waterfall model, the agile model, and the spiral model.

A waterfall model is a traditional software development approach involving a linear process.

The agile model is more flexible and allows for rapid development and iteration.

A spiral model is a hybrid approach that combines elements of the waterfall and agile models.

Each methodology has its strengths and weaknesses, and the best approach for a given project will depend on that project's specific needs and goals.

Key Steps in the Software Development Process

Several critical steps in the software development process include requirements gathering, design, coding, testing, and deployment.

1. Need Identification

Need identification is one of the critical stages in the software development process. Identification is a stage at which the project team works with the client to understand the specific needs and requirements of the software. This information is then used to develop a detailed specification for the software.

2. Requirement Analysis

After a software development project has been initiated, the first step is to perform a requirements analysis. It allows the project team to understand what the customer or user is looking for clearly.

The requirements analysis will involve interviews, surveys, and other research methods to gather the necessary information. Once the data has been collected, it must be analyzed and organized so that the project team can start to develop a plan for the software.

3. Design

Design is a crucial step in the software development process. It is transforming user requirements into a software system that meets those requirements. Design involves choosing suitable data structures, algorithms, and interfaces to implement the system. It also involves trade-offs among performance, memory usage, and other factors. Good design can make a significant difference in the quality and usability of a software system.

4. Development and Implementation

Development and implementation take a software program from its initial conception to its eventual release. Development and implementation are crucial because it ensures that the program meets the user's specific needs, is easy to use, and can be released promptly.

5. Testing

Testing is an essential step in the software development process, and it helps ensure that the software meets all the requirements and functions correctly. Testing also helps identify any errors or bugs in the software so that the team can be fixed before the software is released to the public.

6. Deployment and Maintenance

Deployment and maintenance are a process which include installing the software on a server, configuring the server, and ensuring that the software is running correctly. Additionally, maintenance involves troubleshooting and resolving any issues that arise.

Why Is Software Development Important?

Software development is a process of creating and maintaining software applications. It is a very important part of the information technology industry because it allows businesses to create custom applications that can automate processes and improve efficiency.

Key Features of Effective Software Development

There are many vital features of practical software development. Still, some of the most important include creating a clear and concise requirements document, using a robust and well-tested software development methodology, and having a solid communication plan between all stakeholders.

- Creating a clear and concise requirements document is critical to the success of any software development project. This document should outline all the functionality that is required, as well as any specific constraints or dependencies. With a clear understanding of the project requirements, it is easier to develop a quality software solution.
- Using a robust and well-tested software development methodology is also essential for the success of a project. And plenty of software development methodologies are available, but not all are equally effective. Some more popular and effective methods include Agile, waterfall, and iterative development.
- Having a solid communication plan between all stakeholders or clients is crucial to the success of any software development project. All stakeholders should know the project requirements, the development methodology used, and the project timeline. Good communication will help to ensure that everyone is on the same page and that the project is completed successfully.

Software development Programs

Software development refers to a set of computer science activities dedicated to the process of creating, designing, deploying and supporting software.

Software itself is the set of instructions or programs that tell a computer what to do. It is independent of hardware and makes computers programmable.

Software development is primarily conducted by programmers, software engineers and software developers. These roles interact and overlap, and the dynamics between them vary greatly across development departments and communities.

Programmers, or coders, write source code to program computers for specific tasks like merging databases, processing online orders, routing communications, conducting searches or displaying text and graphics. Programmers typically interpret instructions from software developers and engineers and use programming languages like C++ or Java to carry them out.

Software engineers apply engineering principles to build software and systems to solve problems. They use modeling language and other tools to devise solutions that can often be applied to problems in a general way, as opposed to merely solving for a specific instance or client. Software engineering solutions adhere to the scientific method and must work in the real world, as with bridges or elevators. Their responsibility has grown as products have become increasingly more intelligent with the addition of microprocessors, sensors and software. Not only are more products relying on software for market differentiation, but their software development must be coordinated with the product's mechanical and electrical development work.

Software developers have a less formal role than engineers and can be closely involved with specific project areas — including writing code. At the same time, they drive the overall software development lifecycle — including working across functional teams to transform requirements into features, managing development teams and processes, and conducting software testing and maintenance.

The work of software development isn’t confined to coders or development teams. Professionals such as scientists, device fabricators and hardware makers also create software code even though they are not primarily software developers.

Editor

Editors or text editors are software programs that enable the user to create and edit text files. In the field of programming, the term editor usually refers to source code editors that include many special features for writing and editing code. Notepad, Wordpad are some of the common editors used on Windows OS and vi, emacs, Jed, pico are the editors on UNIX OS. Features normally associated with text editors are — moving the cursor, deleting, replacing, pasting, finding, finding and replacing, saving etc.

Types of Editors

There are generally five types of editors as described below:

1. ***Line editor:*** In this, you can only edit one line at a time or an integral number of lines. You cannot have a free-flowing sequence of characters. It will take care of only one line.

Ex : Teleprinter, edlin, teco

2. ***Stream editors:*** In this type of editors, the file is treated as continuous flow or sequence of characters instead of line numbers, which means here you can type paragraphs.

Ex : Sed editor in UNIX

3. ***Screen editors:*** In this type of editors, the user is able to see the cursor on the screen and can make a copy, cut, paste operation easily. It is very easy to use mouse pointer.

Ex : vi, emacs, Notepad

4. ***Word Processor:*** Overcoming the limitations of screen editors, it allows one to use some format to insert images, files, videos, use font, size, style features. It majorly focuses on Natural language.
5. ***Structure Editor:*** Structure editor focuses on programming languages. It provides features to write and edit source code.

Ex : Netbeans IDE, gEdit.

Editing Process

We all by now understand that editors are the program which is used to create, edit and modify a document. A document may include some images, files, text, equations, and diagrams as well. But we will be limited to text editors only whose main elements are character strings.

The document editing process mainly compromises of the following four tasks:

- The part of the document to edited or modifies is selected.
- Determining how to format this lines on view and how to display it.
- Specify and execute the operations that modify the document.
- Update the view properly.

The above steps include filtering, formatting, and traveling.

Formatting: Visibility on display screen.

Filtering: Finding out the main/important subset.

Traveling: Locating the area of interest.

Translator

- A translator or programming language processor is a computer program that converts the programming instructions written in human convenient form into machine language codes that the computers understand and process.

- It is a generic term that can refer to a compiler, assembler, or interpreter—anything that converts code from one computer language into another.
- These include translations between high-level and human-readable computer languages such as C++ and Java, intermediate-level languages such as Java bytecode, low-level languages such as the assembly language and machine code, and between similar levels of language on different computing platforms, as well as from any of these to any other of these.
- The term is also used for translators between software implementations and hardware/ASIC microchip implementations of the same program, and from software descriptions of a microchip to the logic gates needed to build it.
- Examples of widely used types of computer language translators include interpreters, compilers and decompilers, assemblers and disassemblers.

Linker

- Linker is a program in a system which helps to link an object modules of program into a single object file. It performs the process of linking.
- Linker is also called link editors. Linking is process of collecting and maintaining piece of code and data into a single file.
- Linker also link a particular module into system library. It takes object modules from assembler as input and forms an executable file as output for loader.

- Linking is performed at both compile time, when the source code is translated into machine code and load time, when the program is loaded into memory by the loader.
- Linking is performed at the last step in compiling a program.

Static Linking

- ✓ In static linking, the liner links all modules of a program before its execution begins; it produces a binary program that does not contain any unresolved external references.
- ✓ If statically linked programs use the same module from a library, each program will get a private copy of the module.
- ✓ If many programs that use the module are in execution at the same time, many copies of the module might be present in memory.

Dynamic Linking

- ✓ Dynamic linking is performed during execution of a binary program.
- ✓ The linker is invoked when an unresolved external reference and resumes execution of the program.
- ✓ This arrangement has several benefits concerning use, sharing and updating of library modules.

- ✓ If the module referenced by a program has already been linked to another program that is in execution, a copy of the module would exist in memory. The same copy of the module could be lined to his program as well, thus saving memory.
- ✓ To facilitate dynamic linking, each program is first processed by the static linker.

Loader

Loader is utility program which takes object code as input prepares it for execution and loads the executable code in to the memory. Thus loader is actually responsible for initiating the execution process.

Functions of loader

The loader is responsible for the activities such as allocation, linking, relocation and loading

1) It allocates the space for program in the memory, by calculating the size of the program. This activity is called allocation.

2) It resolves the symbolic references (code/data) between the object modules by assigning all the user subroutine and library subroutine addresses. This activity is called linking.

3) There are some address dependent locations in the program, such address constants must be adjusted according to allocated space, such activity done by loader is called relocation.

4) Finally it places all the machine instructions and data of corresponding programs and subroutines into the memory. Thus program now becomes ready for execution.

Comparison between Linker and Loader:

BASIS FOR COMPARISON	**LINKER**	**LOADER**
Basic	It generates the executable module of a source program.	It loads the executable module to the main memory.
Input	It takes as input, the object code generated by an assembler.	It takes executable module generated by a linker.
Function	It combines all the object modules of a source code to generate an executable module.	It allocates the addresses to an executable module in main memory for execution.
Type/Approach	Linkage Editor, Dynamic linker.	Absolute loading, Relocatable loading and Dynamic Run-time loading.

Types of loader

1. Compile and go to loader
2. General loader
3. absolute loader
4. direct linking loader (DLL)

Debugger

Debugging is the process of identifying and resolving errors, or bugs, in a software system. It is an important aspect of software engineering because bugs can cause a software system to malfunction, and can lead to poor performance or incorrect results. Debugging can be a time-consuming and complex task, but it is essential for ensuring that a software system is functioning correctly.

There are several common methods and techniques used in debugging, including:

1. ***Code Inspection:*** This involves manually reviewing the source code of a software system to identify potential bugs or errors.
2. ***Debugging Tools:*** There are various tools available for debugging such as debuggers, trace tools, and profilers that can be used to identify and resolve bugs.
3. ***Unit Testing:*** This involves testing individual units or components of a software system to identify bugs or errors.

4. ***Integration Testing:*** This involves testing the interactions between different components of a software system to identify bugs or errors.
5. ***System Testing:*** This involves testing the entire software system to identify bugs or errors.
6. ***Monitoring:*** This involves monitoring a software system for unusual behavior or performance issues that can indicate the presence of bugs or errors.
7. ***Logging:*** This involves recording events and messages related to the software system, which can be used to identify bugs or errors.

Database Fundamentals

What is a database?

- A ***database*** is a repository of data, designed to support efficient data storage, retrieval and maintenance.
- Multiple types of databases exist to suit various industry requirements.
- A database may be specialized to store binary files, documents, images, videos, relational data, multidimensional data, transactional data, analytic data, or geographic data to name a few.
- Data can be stored in various forms, namely tabular, hierarchical and graphical forms.
- If data is stored in a tabular form then it is called a ***relational database***.
- When data is organized in a tree structure form, it is called a ***hierarchical database***.
- Data stored as graphs representing relationships between objects is referred to as a network database.

What is a database management system?

- A ***database management system***, or simply ***DBMS***, is a set of software tools that control access, organize, store, manage, retrieve and maintain data in a database.
- In practical use, the terms database, database server, database system, data server, and database management systems are often used interchangeably.

Why do we need database software or a DBMS?

- The answer lies in the way users access the data and the handle of corresponding challenges.
- ***First***, we need the ability to have multiple users insert, update and delete data to the same data file such that, different users will not cause the data to become inconsistent, and no data should be inadvertently lost through these operations.
- We *also* ***need*** to have a standard interface for data access, tools for data backup, data restore and recovery,
- ***And*** a way to handle other challenges such as the capability to work with huge volumes of data and users.
- Database software has been designed to handle all of these challenges.

Information models and data models

- An information model is an abstract, formal representation of entities that includes their properties, relationships and the operations that can be performed on them.
- The primary motivation behind the concept is to formalize the description of a problem domain without constraining how that description will be mapped to an actual implementation in software.
- There may be many mappings of the Information Model. Such mappings are called data models.

- Information Models and Data Models are different because they serve different purposes.
- The main purpose of an Information Model is to model managed objects at a conceptual level, independent of any specific implementations or protocols used to transport the data.
- Data Models, on the other hand, are defined at a more concrete level and include many details. They are intended for software developers and include protocol-specific constructs.
- Figure 1 illustrates the relationship between an Information Model and a Data Model.

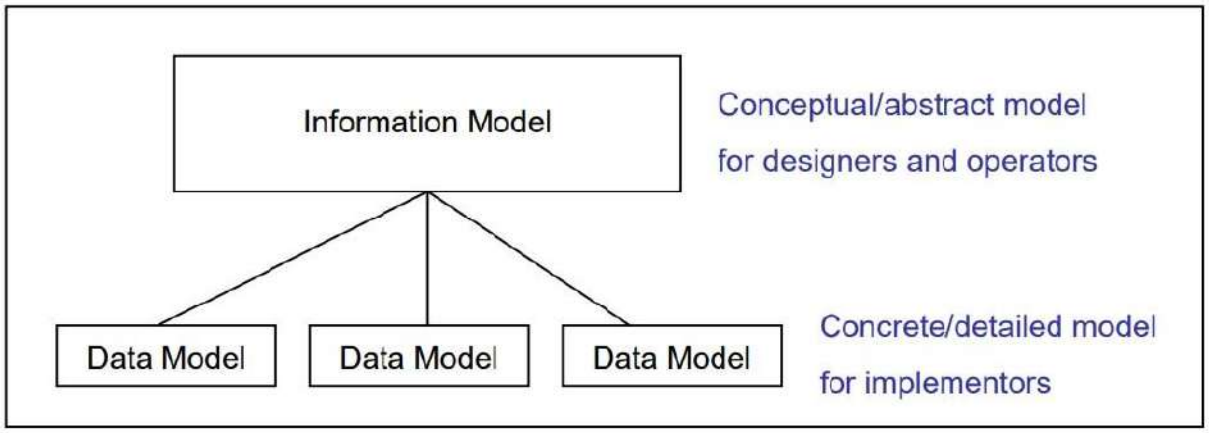


Figure 1 Relationship between an Information Model and a Data Model

Types of information models

There are many types of information models, the main types are:

- Network (CODASYL): 1970’s
- Hierarchical (IMS): late 1960’s and 1970’s

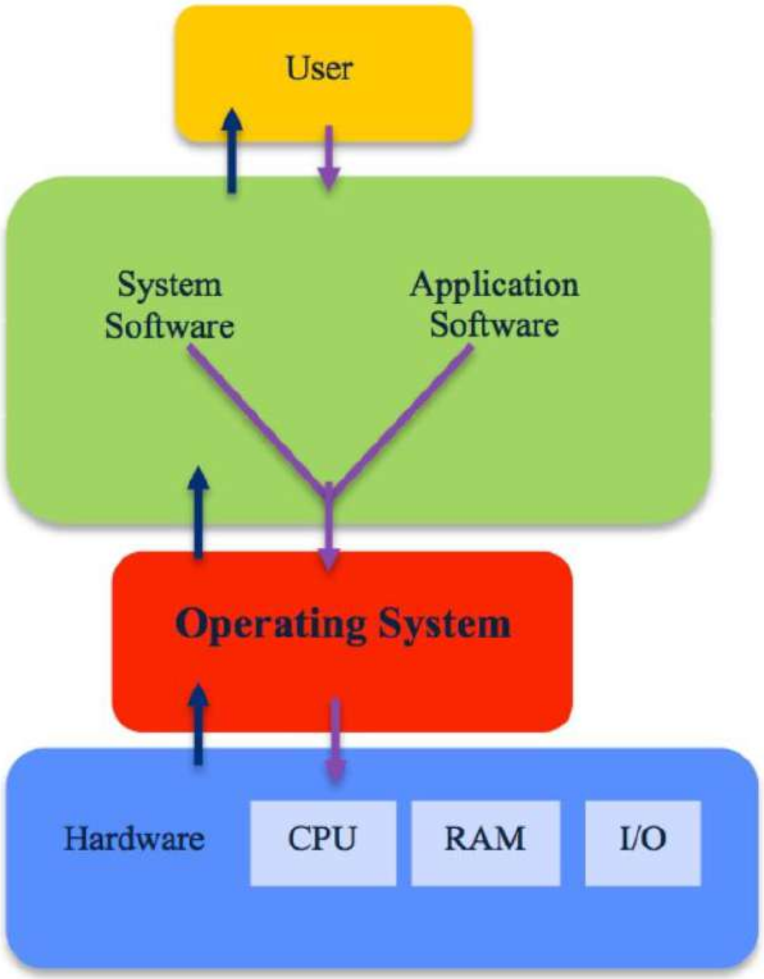
- Relational: 1970’s and early 1980’s
- Entity-Relationship: 1970’s
- Object-relational: late 1980’s and early 1990’s

Operating system basics

An ***operating system*** is a program that acts as an interface between the computer user and computer hardware, and controls the execution of programs.

The operating system’s job

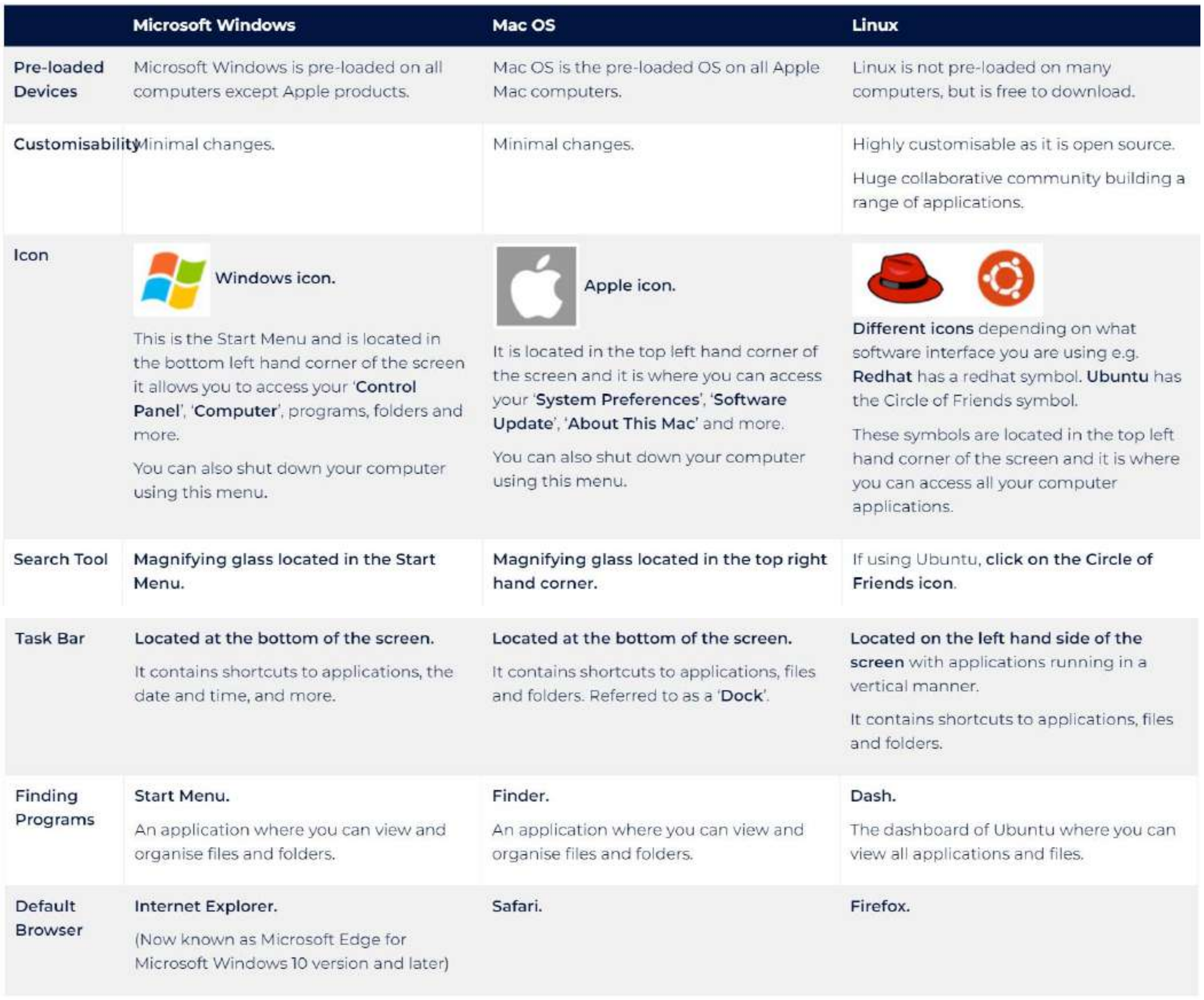
- The operating system (OS) manages all of the software and hardware on the computer.
- It performs basic tasks such as file, memory and process management, handling input and output, and controlling peripheral devices such as disk drives and printers.
- Most of the time, there are several different computer programs running at the same time, and they all need to access your computer’s central processing unit (CPU), memory and storage.
- The operating system coordinates all of this to make sure each program gets what it needs.



- In the image above the User interfaces with the System & Application software.
- The System & Application software interfaces with the Operating System.
- The Operating system interfaces with the Hardware.
- Each of these interfaces are two way transactions with each sending and receiving data.

Types of operating systems

- Operating systems usually come pre-loaded on any computer you buy.
- Most individuals use the operating system that already comes with their computer however it is possible to upgrade or change the initial operating system to suit your preference.
- Different operating systems will work in different ways.
- They may appear visually different, have different terms for common functions and organize programs in different ways.
- There are many operating systems that are available however the three most common operating systems are ***Microsoft's Windows***, ***Apple's macOS*** and ***Linux.***
- In the table below, we will outline a few of the key differences between each system.



	Microsoft Windows	**Mac OS**	**Linux**
Pre-loaded Devices	Microsoft Windows is pre-loaded on all computers except Apple products.	Mac OS is the pre-loaded OS on all Apple Mac computers.	Linux is not pre-loaded on many computers, but is free to download.
Customisability	Minimal changes.	Minimal changes.	Highly customisable as it is open source. Huge collaborative community building a range of applications.
Icon	**Windows icon.** This is the Start Menu and is located in the bottom left hand corner of the screen it allows you to access your '**Control Panel**', '**Computer**', programs, folders and more. You can also shut down your computer using this menu.	**Apple icon.** It is located in the top left hand corner of the screen and it is where you can access your '**System Preferences**', '**Software Update**', '**About This Mac**' and more. You can also shut down your computer using this menu.	**Different icons** depending on what software interface you are using e.g. **Redhat** has a redhat symbol. **Ubuntu** has the Circle of Friends symbol. These symbols are located in the top left hand corner of the screen and it is where you can access all your computer applications.
Search Tool	**Magnifying glass located in the Start Menu.**	**Magnifying glass located in the top right hand corner.**	If using Ubuntu, **click on the Circle of Friends icon**.
Task Bar	**Located at the bottom of the screen.** It contains shortcuts to applications, the date and time, and more.	**Located at the bottom of the screen.** It contains shortcuts to applications, files and folders. Referred to as a '**Dock**'.	**Located on the left hand side of the screen** with applications running in a vertical manner. It contains shortcuts to applications, files and folders.
Finding Programs	**Start Menu.** An application where you can view and organise files and folders.	**Finder.** An application where you can view and organise files and folders.	**Dash.** The dashboard of Ubuntu where you can view all applications and files.
Default Browser	**Internet Explorer.** (Now known as Microsoft Edge for Microsoft Windows 10 version and later)	**Safari.**	**Firefox.**

Software Analysis

- ***Software analysis*** is basically a requirement analysis that aims to determine the tasks that are needed to get fully functional software.
- This analysis undergoes various requirements of stakeholders, documenting, and validating software and system requirements.
- Without requirement analysis, a project will not be completed and would lead to failure as design can only be implemented after its analysis.

There are three types of requirement analysis: –

1. *Eliciting requirements:* Simple collection of data through stakeholders in the form of interviews and business process documentation.
2. *Recording requirements:* The documented requirements are recorded in various forms that involve the use of a summary list.
3. *Analyzing requirements:* Last stage where the gathered resources are clear, concise, unduplicated, valid, and consistent without any conflicts arising.

Software Analysis Tools

There are four basic types of software analysis tools:

- ***Code Coverage***—Measures the amount of the software that has been executed.

- ***Instruction Trace***—creates a record of exactly what happens as the code is executed.
- ***Memory Analysis***—Tracks the code's memory usage and identifies possible errors.
- ***Performance Analysis***—Identifies performance bottlenecks and other issues allowing fine-tuning of the application for higher performance.

Types of Software Analysis

1. *Instrumentation*

This method of software analysis involves inserting markers within lines of code to measure the location and timing of executions.

2. *Code Sampling*

Code sampling involves monitoring code as it runs and inspecting the PC at fixed intervals.

3. *Built-In SoC Capabilities*

System-on-chip (SoC) systems present unique challenges. When a complete networking system is implemented within a single integrated circuit (IC), there are multiple modular components interconnected by a common infrastructure. All the various subsystems and blocks must be designed, verified and completed simultaneously, in parallel fashion. It is now possible to design SoCs with Instruction Trace capabilities, as well as cache monitors and performance timing information.

Flowchart

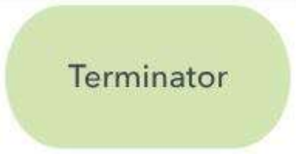
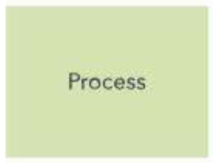
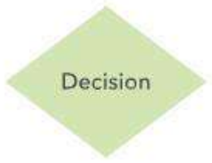
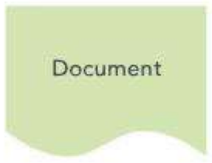
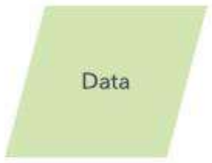
- A ***flowchart*** is a diagram that depicts a process, system or computer algorithm.
- A *flowchart can also be defined as* a diagrammatic representation of an algorithm, a step-by-step approach to solving a task.
- Flowcharts are used to design and document simple processes or programs. Like other types of diagrams, they help visualize the process.
- Flowcharts, sometimes spelled as flow charts, use rectangles, ovals, diamonds and potentially numerous other shapes to define the type of step, along with connecting arrows to define flow and sequence.
- They can range from simple, hand-drawn charts to comprehensive computer-drawn diagrams depicting multiple steps and routes.

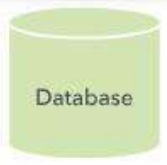
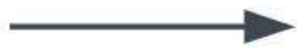
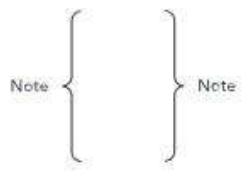
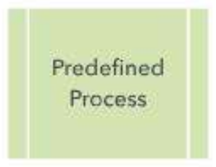
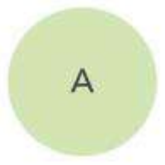
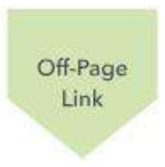
Flowchart symbols

Flowcharts typically use the following main symbols:

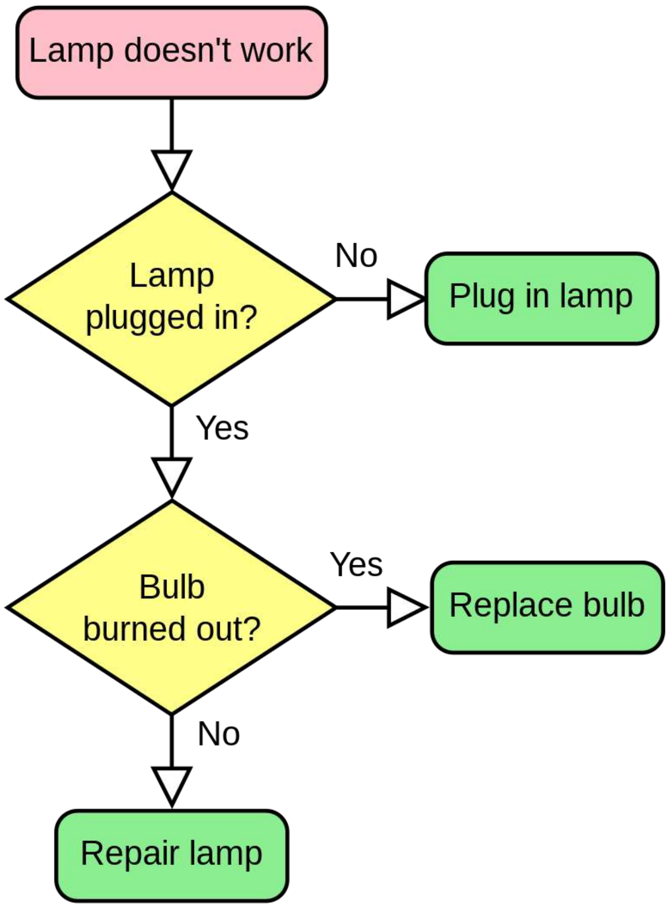
- ✓ A process step, usually called an activity, is denoted by a ***rectangular box***.
- ✓ A decision is usually denoted by a ***diamond***.

Here are some of the common flowchart symbols.

Terminal/Terminator	
Process	
Decision	
Document	
Data, or Input/Output	

Stored Data	
Flow Arrow	
Comment or Annotation	
Predefined process	
On-page connector/reference	
Off-page connector/reference	

Ex. A simple flowchart representing a process for dealing with a non-functioning lamp.



Software Design

- ***Software design*** is the process of envisioning and defining software solutions to one or more sets of problems.
- One of the main components of software design is the software requirements analysis (SRA).
- SRA is a part of the software development process that lists specifications used in software engineering.
- Depending on the environment, the design often varies, whether it is created from reliable frameworks or implemented with suitable design patterns.
- Design examples include operation systems, webpages, mobile devices or even the new cloud computing paradigm.

Difference between software analysis and design.

- ✓ The main difference between software analysis and design is that the output of a software analysis consists of smaller problems to solve.
- ✓ Additionally, the analysis should not be designed very differently across different team members or groups.
- ✓ In contrast, the design focuses on capabilities, and thus multiple designs for the same problem can and will exist.

Design concepts

The design concepts provide the software designer with a foundation from which more sophisticated methods can be applied. A set of fundamental design concepts has evolved. They are as follows:

1. ***Abstraction*** - Abstraction is the process or result of generalization by reducing the information content of a concept or an observable phenomenon, typically in order to retain only information which is relevant for a particular purpose. It is an act of representing essential features without including the background details or explanations.
2. ***Refinement*** - It is the process of elaboration. A hierarchy is developed by decomposing a macroscopic statement of function in a step-wise fashion until programming language statements are reached. In each step, one or several instructions of a given program are decomposed into more detailed instructions. Abstraction and Refinement are complementary concepts.
3. ***Modularity*** - Software architecture is divided into components called modules.
4. ***Software Architecture*** - It refers to the overall structure of the software and the ways in which that structure provides conceptual integrity for a system. Good software architecture will yield a good return on investment with respect to the desired outcome of the project, e.g. in terms of performance, quality, schedule and cost.
5. ***Control Hierarchy*** - A program structure that represents the organization of a program component and implies a hierarchy of control.

6. ***Structural Partitioning*** - The program structure can be divided horizontally and vertically. Horizontal partitions define separate branches of modular hierarchy for each major program function. Vertical partitioning suggests that control and work should be distributed top down in the program structure.
7. ***Data Structure*** - It is a representation of the logical relationship among individual elements of data.
8. ***Software Procedure*** - It focuses on the processing of each module individually.
9. ***Information Hiding*** - Modules should be specified and designed so that information contained within a module is inaccessible to other modules that have no need for such information.

Design considerations

There are many aspects to consider in the design of a piece of software. The importance of each consideration should reflect the goals and expectations that the software is being created to meet. Some of these aspects are:

Compatibility - The software is able to operate with other products that are designed for interoperability with another product.

Extensibility - New capabilities can be added to the software without major changes to the underlying architecture.

Modularity - the resulting software comprises well defined, independent components which leads to better maintainability.

Fault-tolerance - The software is resistant to and able to recover from component failure.

Maintainability - A measure of how easily bug fixes or functional modifications can be accomplished.

Reliability (Software durability) - The software is able to perform a required function under stated conditions for a specified period of time.

Reusability - The ability to use some or all of the aspects of the preexisting software in other projects with little to no modification.

Robustness - The software is able to operate under stress or tolerate unpredictable or invalid input.

Security - The software is able to withstand and resist hostile acts and influences.

Usability - The software user interface must be usable for its target user/audience.
Performance - The software performs its tasks within a time-frame that is acceptable for the user, and does not require too much memory.

Portability - The software should be usable across a number of different conditions and environments.

Scalability - The software adapts well to increasing data or added features or number of users.

Data flow diagram (DFD)

- A data flow diagram (DFD) maps out the flow of information for any process or system.
- It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination.
- Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled.
- They can be used to analyze an existing system or model a new one.

Components of DFD

The Data Flow Diagram has 4 components:

Process Input to output transformation in a system takes place because of process function. The symbols of a process are rectangular with rounded corners, oval, rectangle or a circle. The process is named a short sentence, in one word or a phrase to express its essence

Data Flow Data flow describes the information transferring between different parts of the systems. The arrow symbol is the symbol of data flow.

Warehouse The data is stored in the warehouse for later use. Two horizontal lines represent the symbol of the store.

Terminator The Terminator is an external entity that stands outside of the system and communicates with the system.

Symbols and Notations Used in DFDs

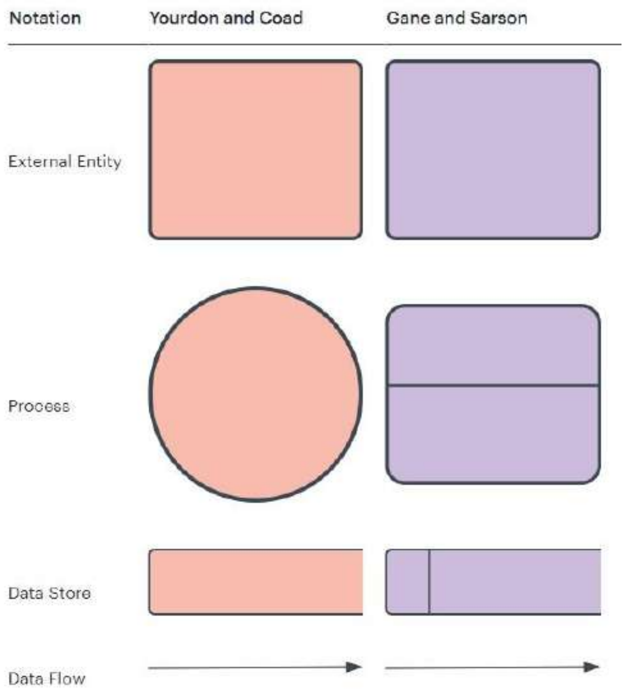
Three common systems of symbols are named after their creators:

Yourdon and Coad

Yourdon and DeMarco

Gane and Sarson

One main difference in their symbols is that Yourdon-Coad and Yourdon-DeMarco use circles for processes, while Gane and Sarson use rectangles with rounded corners, sometimes called lozenges.



DFD rules and tips

- ✓ Each process should have at least one input and an output.
- ✓ Each data store should have at least one data flow in and one data flow out.
- ✓ Data stored in a system must go through a process.
- ✓ All processes in a DFD go to another process or a data store.

Walkthrough

- In software engineering, a ***walkthrough*** or walk-through is a form of software peer review "in which a designer or programmer leads members of the development team and other interested parties through a software product, and the participants ask questions and make comments about possible errors, violation of development standards, and other problems".
- The reviews are also performed by assessors, specialists, etc. and are suggested or mandatory as required by norms and standards.

Objectives and participants

In general, a walkthrough has one or two broad objectives:

- ✓ to gain feedback about the technical quality or content of the document;
- ✓ and/or to familiarize the audience with the content.

Safety and Maintenance

Keeping Your Computer Clean

- Keeping Your Computer Clean is represented by physical cleaning.
- Dust isn't just unattractive—it can potentially damage or even destroy parts of your computer.
- Cleaning your computer regularly will help you keep it working properly and avoid expensive repairs.
- Physical cleaning means cleaning all computer’s parts which represented by: mouse, keyboard and monitor.

Protecting Your Computer

- Today we use internet-connected devices in all aspects of our lives.
- We go online to search for information, shop, bank, do homework, play games, and stay in touch with family and friends through social networking.
- As a result, our devices contain a wealth of personal information about us. This may include banking and other financial records, and medical information—information that we want to protect.
- If your devices are not protected, identity thieves and other fraudsters may be able to get access and steal your personal information.
- Spammers could use your computer as a "zombie drone" to send spam that looks like it came from you.

- Malicious viruses or spyware could be deposited on your computer, slowing it down or destroying files.
- The following tips are offered to help you lower your risk while you're online:

➢ *Keep your device secure*

Make sure to download recommended updates from your device's manufacturer or operating system provider, especially for important software such as your internet browser.

➢ Keep up-to-date

Update your system, browser, and important apps regularly, taking advantage of automatic updating when it's available.

➢ Antivirus software

Antivirus software protects your device from viruses that can destroy your data, slow down or crash your device, or allow spammers to send email through your account.

➢ Antispyware software

Spyware is software installed without your knowledge or consent that can monitor your online activities and collect personal information while you're online. Some kinds of spyware, called keyloggers, record everything you key in—including your passwords and financial information.

Spyware protection is included in some antivirus software programs.

- Firewalls

A firewall is a software program or piece of hardware that blocks hackers from entering and using your computer. Hackers search the internet the way some telemarketers automatically dial random phone numbers. They send out pings (calls) to thousands of computers and wait for responses. Firewalls prevent your computer from responding to these random calls. A firewall blocks communications to and from sources you don't permit. This is especially important if you have a high-speed internet connection, like DSL or cable.

Use strong protection

Making use of complex passwords and strong methods of authentication can help keep your personal information secure.

- ✓ Choose strong passwords
- ✓ Use stronger authentication

Protect your private information

While checking email, visiting websites, posting to social media, or shopping, pay attention to where you click and who you give your information to. Unscrupulous websites or data thieves can attempt to trick you into giving them your personal data.

- ✓ Be careful what you click
- ✓ Shop safely

- ✓ Be careful what you share
- ✓ Responding to data breaches

Creating a Safe Workspace

- In addition to keeping your computer healthy, it's important to think about your own health.
- Using a computer involves a lot of repetitive motions such as typing and using the mouse.
- Over time, these motions can begin to negatively impact your body, especially your wrists, neck, and back.
- Staring at a monitor for long periods of time can also cause eye strain.
- To minimize these risks, you should take a few moments to make sure your workspace is arranged in a comfortable and healthy way.

Avoiding strain and injury

- Computer ergonomics is the science of equipment design and how specific equipment usage and placement can reduce a user's discomfort and increase productivity.
- Some equipment is designed with special attention to ergonomics, like ergonomic keyboards and ergonomic chairs.
- Here are a few tips to help you avoid injury in your workspace:

- ✓ ***Adjust your chair:*** Make sure your chair is adjusted to allow you to sit in a natural, comfortable position.
- ✓ ***Keep the keyboard at a comfortable height:*** Try to place the keyboard in a position that allows you to keep your wrists straight and relaxed to avoid wrist strain.
- ✓ ***Keep the mouse close to the keyboard:*** If possible, place the mouse right next to the keyboard. If the mouse is too far away, it may be uncomfortable or awkward to reach for it.
- ✓ ***Place the monitor at a comfortable distance:*** The ideal position for a monitor is 20 to 40 inches away from your eyes. It should also be at eye level or slightly lower.
- ✓ ***Avoid clutter:*** The computer area can quickly become cluttered with paper, computer accessories, and other items. By keeping this area as uncluttered as possible, you can improve your productivity and prevent strain and injury.
- ✓ ***Take frequent breaks:*** It's important to take breaks while you're working at your computer. To avoid eye strain, you should look away from the monitor every once in a while. You can also stand up and walk around to avoid sitting in the same position for long periods of time. Programs such as *Workrave* can automatically remind you to take breaks.