Chapter 3

Software

Chapter 3

SOFTWARE - TOPIC

- System software
  - Features of System Software
  - Type of System Software
    - operating system
    - Utilities

- Application software
  - The need for application software
  - Type of application software
  - Advantage and disadvantage of application software
  - Different between System Software and Application Software

- Programming software
  - History of Programming software
  - Type of programming software
  - Component of programming software
  - Example of programming software

- Malicious software

Introduction

- Computers and computer applications have become a part of the fabric of everyday life
  - They are great as long as they are working
- We give little thought to the processes and programs running behind the scenes to keep them functioning effectively.
- Such programs are:
  - System software
  - Application software
  - Programming software
  - Malicious software

System software

System Software

- System software is a program designed to run a computer's hardware and applications and manage its resources :
  - memory
  - processors
  - devices.
- It also provides a platform for running application software,
- system software is typically bundled with a computer's operating system.

Copyright © McGraw-Hill Education. Permission required for reproduction or display.

(left): © Image Source, all rights reserved, RF; (middle top): © McGraw-Hill Education/Aaron Roeth, photographer;
(middle bottom): Used with permission from Microsoft; (top right, middle right, bottom right): © Maxim Kazmin/Getty Images RF

System Software - Features of System Software

1. High Speed - high speed is an essential feature of system software
2. Versatile - is determined by how many different programs it can run simultaneously.
3. Hard to Manipulate - is determined by how difficult it is to change settings and configurations.
4. Close to the System - The closeness of system software to the System determines its accessibility.

System Software - Operating Systems

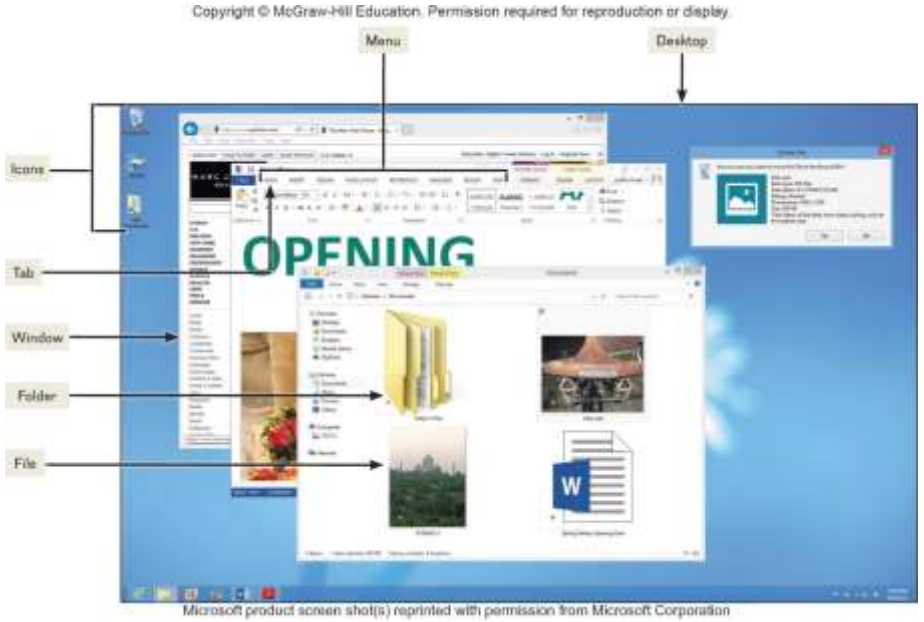
A collection of programs that handle technical tasks

- Functions
  - Manages computer resources
  - Coordinate memory, processing, storage, printers and monitors
  - Monitor system performance
  - Schedule tasks
  - Provide security
  - Start-up the computer
- Provides user interface
  - Graphical user interface (GUI)
- Runs applications
  - Multitasking
  - Foreground and background applications

System Software - Operating Systems

Features of an Operating System

- Booting – starting or restarting the computer
- Features in common with application software
  - Icons
  - Pointer
  - Windows
  - Menus
  - Tabs
  - Dialog boxes
  - Help
  - Gesture Control
- Files and Folders
  - Files share data and programs
  - Folders store related files



System Software - Operating Systems

Categories of Operating Systems

- Three basic categories
  - Embedded operating systems – RTOS (real-time operating systems)
    - Smartphones
    - Smartwatches
    - Video game systems
  - Stand-alone operating systems
    - Also called desktop operating system
  - Network operating systems (linked computers)
    - Windows Server, Linux, Unix
    - OS stored on network server which coordinates all communication between the other computers

Copyright © McGraw-Hill Education. Permission required for reproduction or display.

© Lenovo, used with permission

System Software - Operating Systems

Mobile Operating Systems

- Mobile OS
  - Embedded operating system
    - Less complicated and more specialized for wireless
  - Some of the best known
    - Android
    - iOS
    - Windows Phone

Copyright © McGraw-Hill Education. Permission required for reproduction or display.

© Stanca Sanda/Alamy

System Software - Operating Systems

Desktop Operating Systems

- Operating systems commonly used by individuals
  - Windows – most widely used
  - Mac OS – powerful and easy to use
  - UNIX – network; originally designed for Web
  - LINUX – non proprietary; free from the Web

Android

Windows 10

Microsoft Windows

Ubuntu

Linux

macOS

Chrome OS

System Software - Operating Systems

Windows - Most Used OS

- **Windows 10**
- Merges Windows desktop and mobile operating systems

Copyright © McGraw-Hill Education. Permission required for reproduction or display.

Used with permission from Microsoft

- **Windows 11**
- Adds haptics to the digital pen
- Can input information not only by typing and clicking, but also through touching, writing, and speaking (voice)

Source: https://www.minitool.com/news/windows-11-vs-windows-10.html

System Software - Operating Systems

Mac OS

- Two most recent versions:
  - Big Sur
    - The most advanced desktop operating system
  - Catalina
    - Next generation OS
- Mac OS X runs on Apple computers:
  - OS X El Capitan
    - Focused performance, stability, and security
  - OS X Yosemite
    - New user interface
  - OS X Mavericks
    - Improved power management

App icon in Launchpad on Mac OS Catalina (left) and Big Sur (right)

Source: https://www.idownloadblog.com/2020/07/07/macos-catalina-big-sur-visual-comparison/

The OS X El Capitan Desktop

Source: https://news.softpedia.com/news/os-x-10-11-el-capitan-review-493129.shtml

System Software - Operating Systems

Mac OS

- The era of Mac OS X is over
- Replaced with new Mac OS

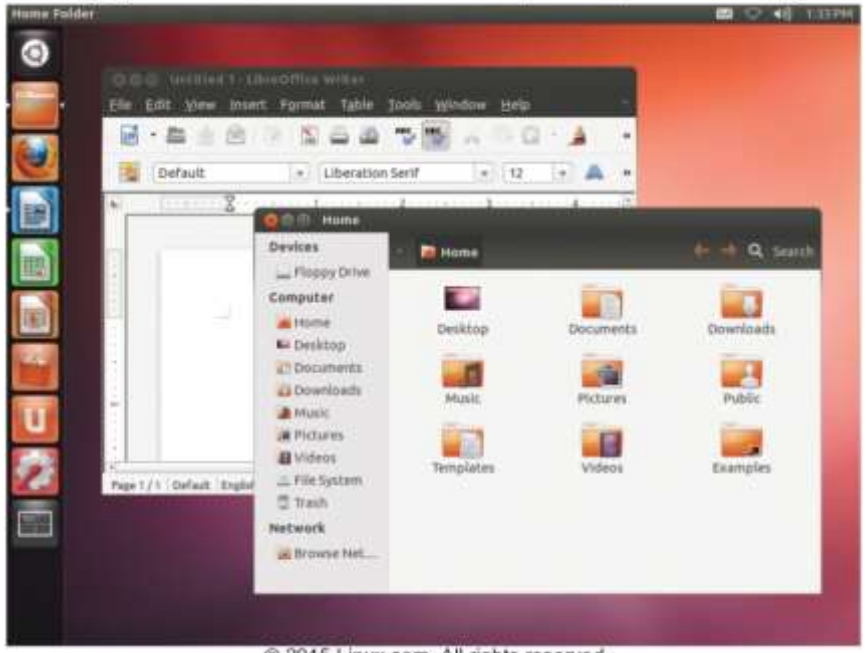
Source: https://arstechnica.com/gadgets/2020/11/macos-11-0-big-sur-the-ars-technica-review/

System Software - Operating Systems

UNIX and LINUX

- UNIX operating system
  - Servers on the Web
  - Mainframe computers
  - Personal Computers
- LINUX - version of UNIX
  - Alternative to windows
  - Open source - free
  - Google Chrome OS based on Linux
    - Focuses on Internet connectivity and cloud computing
    - Speed is determined by the speed of the Internet

Copyright © McGraw-Hill Education. Permission required for reproduction or display.



© 2015 Linux.com. All rights reserved

Virtualization

- Ability to support multiple operating systems on a single physical machine
- Software virtualization
  - Each virtual machine appears as a separate independent computer
    - Host operating system
    - Guest operating system
- Parallels
  - Mac to run Windows programs in Mac OS and Mac OS X

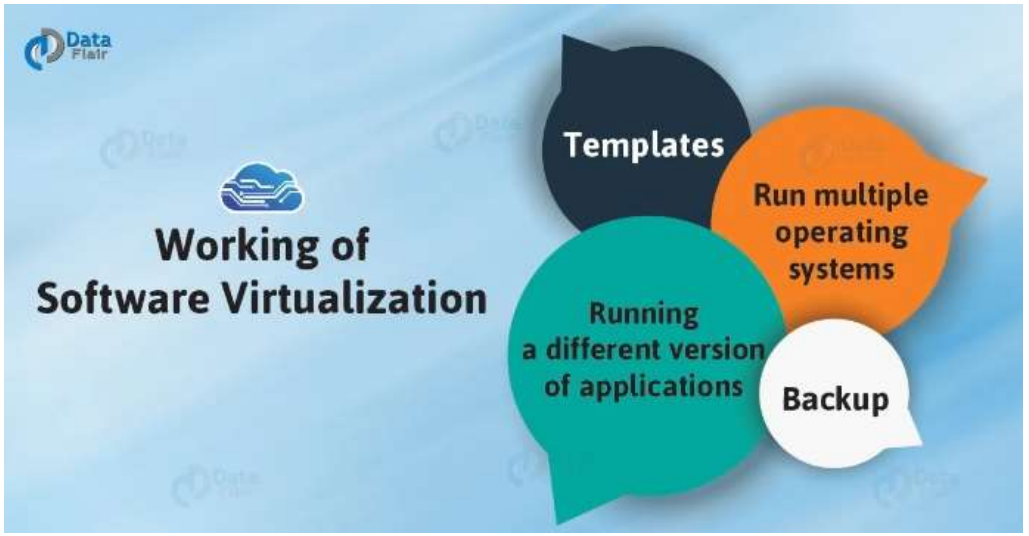
Copyright © McGraw-Hill Education. Permission required for reproduction or display.

Microsoft product screen shot(s) reprinted with permission from Microsoft Corporation

System Software - Operating Systems

Types of Software Virtualization

- Operating System Virtualization
  - A hardware is used which consists of software on which different operating systems work
- Application Virtualization
  - Refer to running an application on a thin client
- Service Virtualization
  - A technique to simulate the behaviour of some components in a mixture of component-based applications



Source: https://data-flair.training/blogs/software-virtualization/

System Software - Utilities

- Specialized programs to make computing easier
- Most essential utilities
  - Troubleshooting or diagnostic programs
    - Recognizes and correct problems
  - Antivirus programs
    - Guard your computer against viruses
  - Backup programs
    - Copies of files to restore if necessary
  - File compression programs
    - Reduces the size of files for more efficient storage

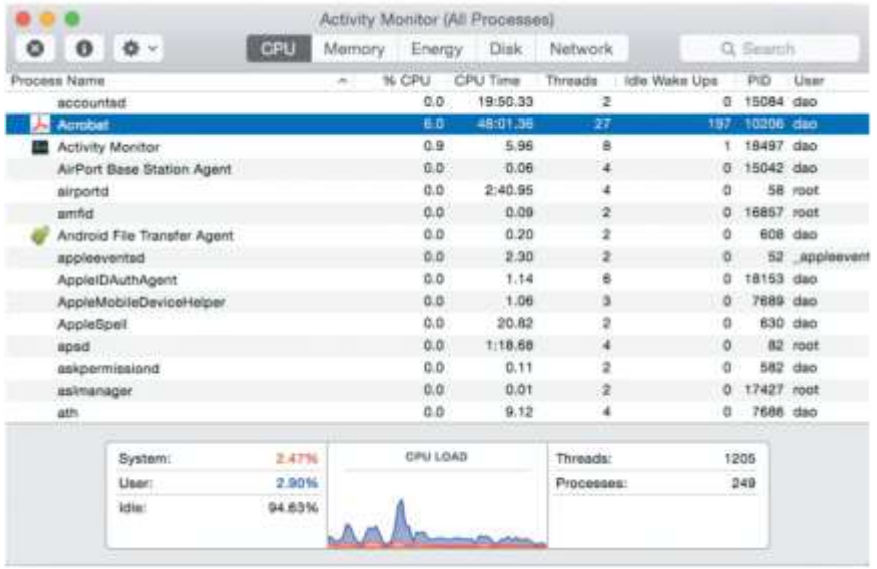
System Software - Utilities

Making IT Work for You – Mac OS X Activity Monitor

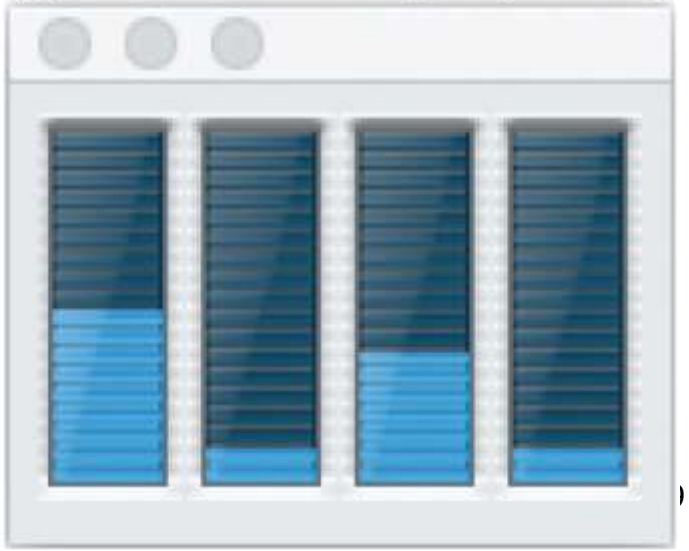
- Has your computer ever just stopped responding? What do you do?
- Mac OS X Activity Monitor is designed to help.

Copyright © McGraw-Hill Education. Permission required for reproduction or display.

Copyright © McGraw-Hill Education. Permission required for reproduction or display.



Copyright © McGraw-Hill Education. Permission required for reproduction or display.



System Software - Utilities

Windows Utilities

Windows Operating Systems includes utilities such as:

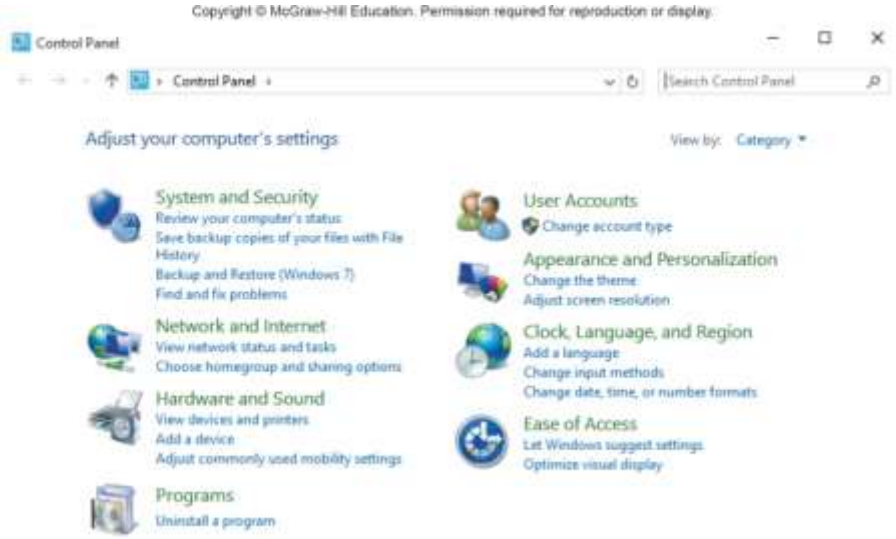
- File History
  - Can create a backup for your hard drive
- Disk Cleanup
  - Identifies and eliminates non essential files
- Disk Defragmenter
  - Rearranges files and unused disk space to optimize performance

System Software - Utilities

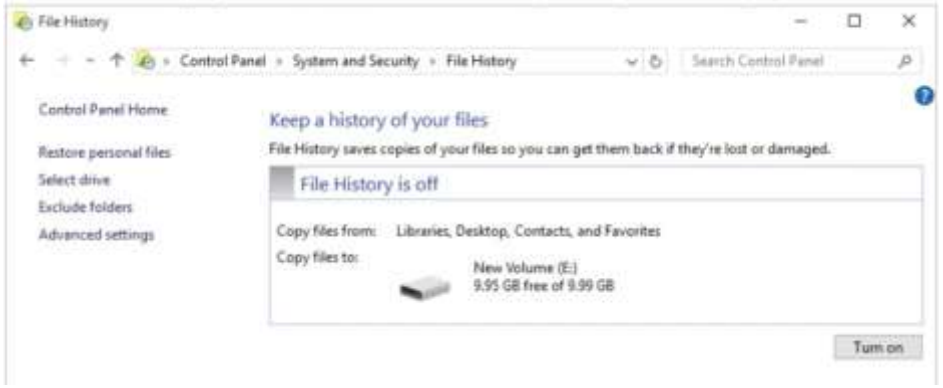
File History

- Utility program included with Windows 8
- Makes a copy of all files in the libraries, contacts, favorites and the desktop
- Helps prevent the effect of disk failure

Copyright © McGraw-Hill Education. Permission required for reproduction or display.



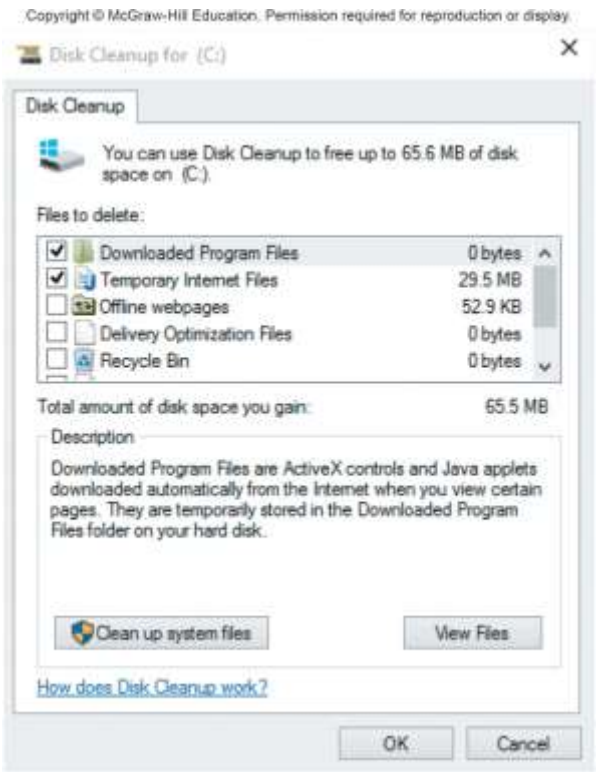
Copyright © McGraw-Hill Education. Permission required for reproduction or display.

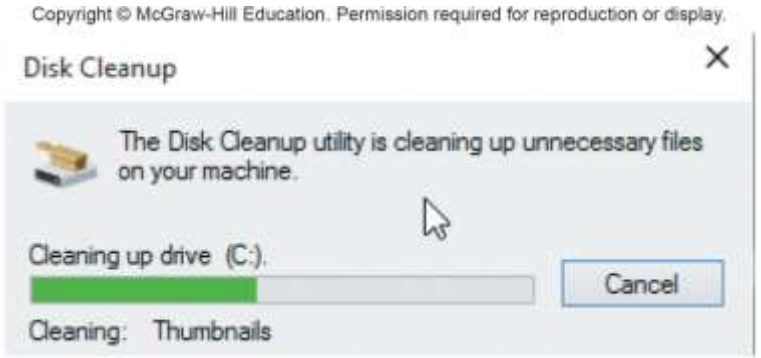


System Software - Utilities

Disk Cleanup

- Identifies and eliminates nonessential files
- Frees up valuable space and improves system performance



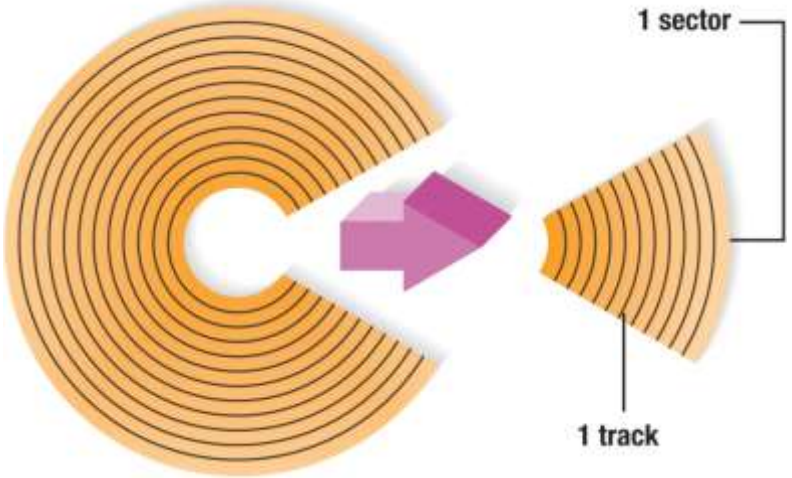


System Software - Utilities

Disk Defragmenter

- Files are organized in tracks and sectors
  - Tracks – concentric ring
  - Sectors – wedge-shaped sections of a track
- Optimize drives - utility program
  - Identifies and eliminates unnecessary fragments
  - Files become fragmented
    - Broken up and stored in non contiguous space
- Rearranges files and unused disk space to optimize operations

Copyright © McGraw-Hill Education. Permission required for reproduction or display.



System Software - Utilities

Utility Suites

- Combine several programs into one package
- Less expensive
- Popular suites
  - Bit Defender
  - Norton Utilities
  - Kaspersky

Copyright © McGraw-Hill Education. Permission required for reproduction or display.

Copyright © 2015 Symantec Corporation. All rights reserved

Application software

Application software

- Application software (App) is a software that performs specific functions for the end user by interacting directly with it.
- The sole purpose of application software is to aid the user in doing specified tasks.

- Example of Application Function:
- Web browsers: Firefox and Google Chrome,
- Personal computer or laptop: Microsoft Word, Excel, and many more
- Smartphone apps: WhatsApp and Telegram,
- Games apps: Candy Crush Saga and many more
- Popular services that people rely on every day: weather or transportation information,
- Apps that connect end users with their businesses.

Application software – The need for application software

End-users can use application software to perform any type of single or many tasks. Following are a few reasons you would definitely need application software (App) on your computer:

- Helps the user in completing specific tasks: Any application software should, in general, be created with the end user in mind. They will primarily benefit the end-user by assisting them in doing specialized activities in a range of industries, including education, business, and entertainment.
- Manages and manipulates data: Application software is used in businesses to manage and alter employee, customer, and other databases.
- Allows users to effectively organize information: Data can be created and handled efficiently by individual users using application software.

Application software – The need for application software

- **Software for healthcare applications for seamless connectivity and quick response**
  - Example:
    - Electronic Health Records (EHRs)
    - Electronic Prescription Systems (EPSs)
    - Telemedicine Software
- **Managing information and data in an organization**
  - Example:
    - watsonx.data
    - IBM Db2
    - Zendesk
- **Managing and checking documents across**
  - Example:
    - MasterControl
    - GLOBODOX
    - Zapier

Application software – The need for application software

- **Software for educational purposes such as Learning Management Systems and e-learning systems**
  - Example:
    - Docebo
    - Adobe Captivate
    - Google Classroom
- **Development of visuals and video for presentation purposes**
  - Example:
    - PowerPoint
    - Powtoon
    - Visme
- **Emails, text messaging, audio and video conferencing involved in presentation**
  - Example
    - Google Meet
    - Webex

Application software –
Types of Application Software

- Word Processing Software

Word Processing refers to using a personal computer (PC) or laptop to create, edit, save and print documents, which can be performed only with specialised software known as a Word Processor.

Example: Microsoft Word

- Spreadsheet Software

Spreadsheet software is a type of computer program that enables a user to perform numerical functions and explore numbers through an automated version of an accounting worksheet.

Example: Microsoft Excel.

- Presentation Software

Presentation software, also commonly known as presentation graphics, is a particular category of application program used to construct sequences of words and a series of pictures that tell a story or help support a speech or public presentation of any information or a launch of new products or services.

Example: PowerPoint

Application software –
Types of Application Software

- **Multimedia Software**

Multimedia software can be described as the combination of text, audio, images, animation, or video to produce a wide scope of interactive content for both professional and personal use.

Example: PowerDvD

- **Web Browsers**

A web browser can take you all over the internet. It retrieves data from other parts of the web and shows it on your desktop or mobile device for viewing. The data is transmitted using the Hypertext Transfer Protocol, which describes how text, images, and video are shared on the World Wide Web.

Example: Google Chrome

- **Educational Software**

Educational software refers to any computer software designed solely for educational reasons. It includes a wide range of software, including language learning software, classroom management software (CMS), and reference software for students and other professionals.

Example: Moodle

Application software –
Types of Application Software

- **Graphics Software**

Graphics software can be reworked with bitmap and/or vector graphics and can be utilised to create label templates.

Example: Canva, Adobe Illustrator, Photoshop

- **Freeware**

Freeware is typically marketed for profit but might be allocated specifically for a business or commercial purpose with the aim of expanding the market share of any newly launched premium product.

Examples: Adobe Reader, Free Studio, and Skype.

- **Shareware**

Shareware is software supplied for free on a trial basis for the user to test or use the programme for a specific amount of days with the understanding that the user may need or want to pay for it later if they are satisfied with the product usage. Some software manufacturers provide a shareware edition of their product with an expiration date built in, such that after 30 days, the user or customer can no longer access the application for further use.

Example: WinZip

Application software –
Types of Application Software

- **Simulation Software**

Simulation software authorises engineers to evaluate, optimise, and compare product designs with other similar software by modelling real-world events in a computer-generated environment.

Example: Simulink, Autodesk Inventor

- **Open Source**

Open source software is a specific code designed to be publicly accessible so that anyone can see, modify, and distribute the code as they see, which fits the purpose. It is designed to be decentralised and coordinated, depending on peer assessment and community production.

Example: GIMP, Shotcut

- **Closed Source**

Closed source software is where the source code is not freely accessible. It is developed and delivered to the customer as a fully compiled, executable set of files. The developer often provides aid to users after purchase and ensures that the software works as the creator foresaw.

Example: Microsoft Windows

Application software – Advantages of Application Software

- It meets every client's particular requirements and needs. The client usually recognizes that they must utilize only one explicit program to finish the task because it is designed explicitly for one reason.
- Businesses that are related to specific applications can restrict access and think about strategies to monitor their activities.
- Standard updates from engineers for Licensed application programming can be obtained using general logic of health.

Application software – Disadvantages of Application Software

- Developing and evolving any application software to achieve specific goals can be quite pricey for developers but this can have a strong impact on their financial plan and income stream, particularly if an unreasonable high amount of time is spent on a product that is not normally worthy.

- Application software that is widely used by many of us and then disseminated on the internet is inherently vulnerable to infection by a bug or other malicious initiatives.

Difference Between System Software and Application Software

System Software

System Software	Application Software
The main purpose of this software is to manage the resources available in the system. It serves as an effective forum for the execution of application software	Application software designed to achieve a certain set of tasks.
System software is documented in a low-level programming language like machine code or assembly language.	Application software is composed in a high-level language like Java, C++,.Net, or PHP.
Usually, when the computer is switched on, system software begins to run and stops when the computer is switched off.	When a user requests, application software runs according to the task it is assigned.
Without system software, a computer system cannot even activate.	User-specific application software is definitely not required to run the system.
The system software has a wide scope of capabilities.	The objective of the application software is to achieve or perform a certain task.
System software comprises language processors (interpreters, compilers, and assemblers), operating	Payroll software, accounting software, MS Office, and so

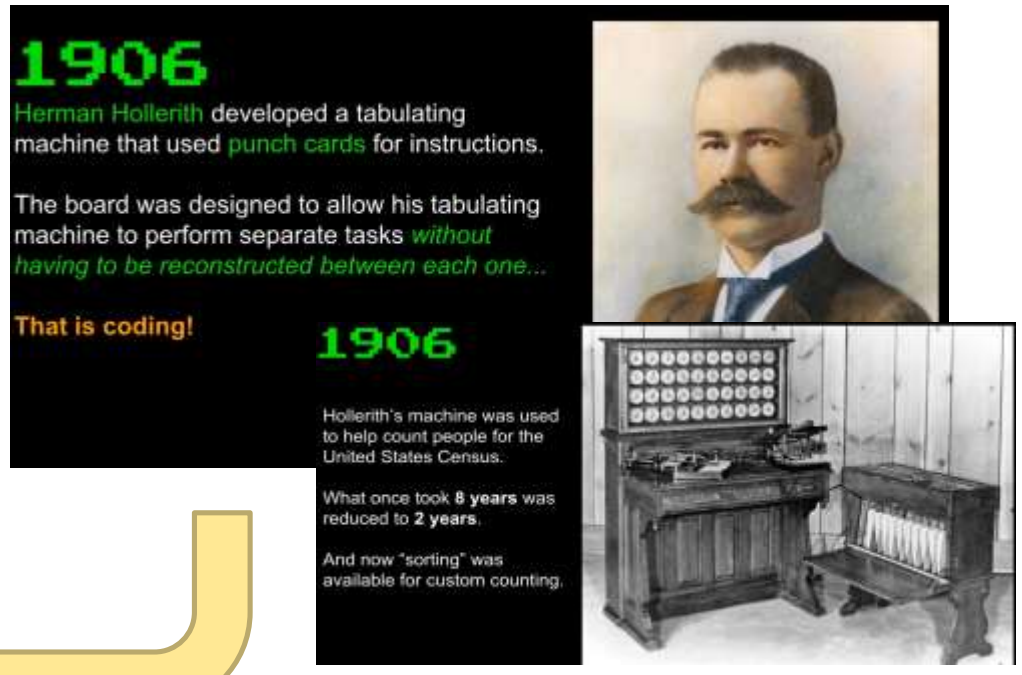
Sources: https://www.geeksforgeeks.org/what-is-application-software/

Programming software

Programming software

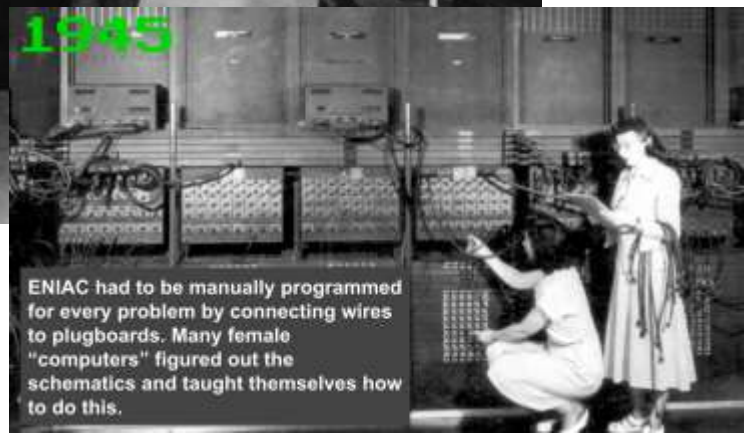
- Programming software is the software which helps the programmer develop other software.
- It is a tool for creating computer code that allows computer software to operate.
- Programming software includes compilers, assemblers, debuggers, interpreters, etc.

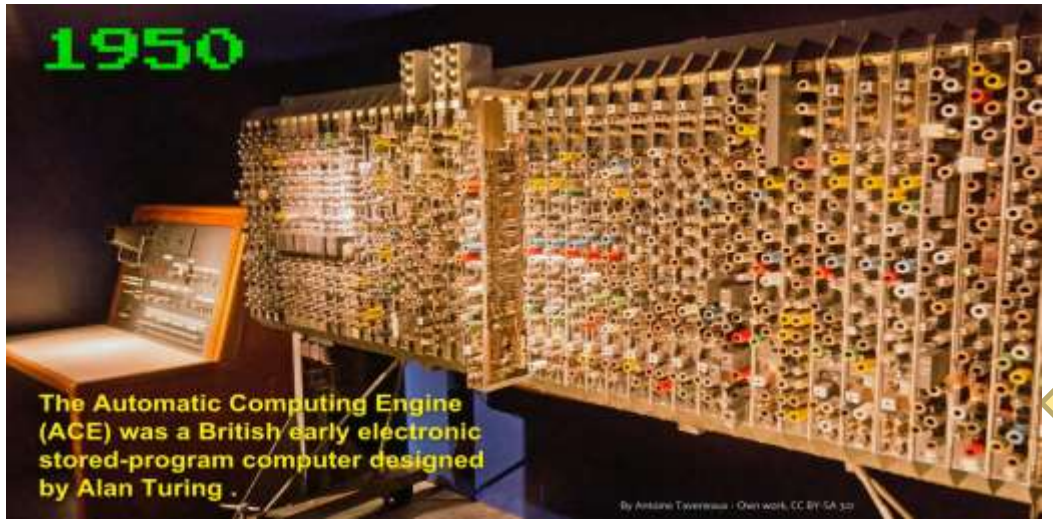
Programming software – History of Programming software



Source: https://youtu.be/M4d3FXu9-3I?feature=shared

Programming software – History of Programming software





Source: https://youtu.be/M4d3FXu9-3I?feature=shared

Programming software – History of Programming software

1954

IBM created America's first high level programming language when a team led by John Backus created **FORTRAN** (FORmula TRANslation).

FORTRAN was originally developed for scientific and engineering programs. It is still used a bit today.

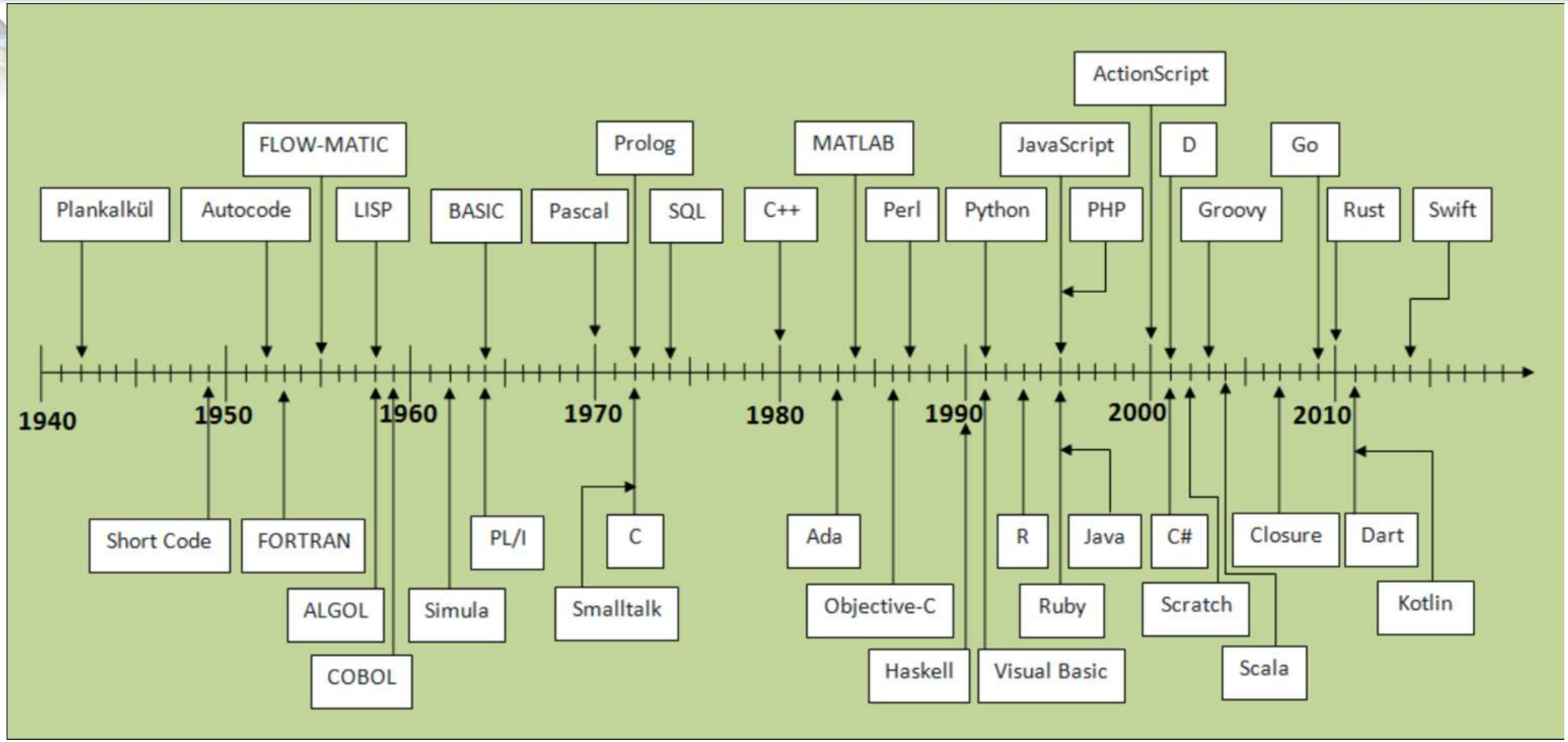
This programming language created the foundation to higher level languages such as HTML and C++.

1959

From 1959 to 1961, Dr. Grace Hopper lead the team that created **COBOL** (Common Business-Oriented Language), the first user-friendly business computer software program.

Source: https://youtu.be/M4d3FXu9-3I?feature=shared

Programming software – History of Programming software



Sources: https://javaconceptoftheday.com/history-of-programming-languages/

Programming software – Type of programming software

All programming languages fall into two broad categories: high-level and low-level.

- High-level Programming Software
  - High-level programming software is far from the code that translates commands on a computer system.
  - It is user-oriented because it has been designed to make it straightforward for a programmer to convert an algorithm into program code.

    Examples: C#, Java, and Python.

- Low-level Programming Software
  - Low-level programming software contains basic instructions for a computer to understand. It involves little-to-no abstraction from the core coding process.

    Examples: assembly and machine code.

Programming software – Component of programming software

- **Compilers**
  - A compiler quickly reads the entire source program in the high-level language and translates it into an equivalent program in machine language.

- **assemblers,**
  - The Assembler converts an Assembly language program into machine code. The source program is an assembler input that contains assembly language instructions. The assembler's output is computer-readable object code or machine code.

- **debuggers,**
  - As the name suggests, the debugger is a used for removing bugs from code. It simply allows other programs to be tested and debugged. Debugger may also provide two operation modes, complete and partial simulation. It is used to prevent software or systems from malfunctioning.

Programming software – Component of programming software

- **interpreters**
  - A language processor converts a single statement from the source program into machine code. An interpreter is a program that executes code immediately before moving on to the following line. If a statement contains an error, the interpreter stops translating at that point and displays an error message.

- **Integrated development environments (IDEs)**

  An integrated development environment (IDE) is a software application that helps programmers develop software code efficiently. It increases developer productivity by combining capabilities such as software editing, building, testing, and packaging in an easy-to-use application. Just as writers use text editors and accountants use spreadsheets, software developers use IDEs to make their job easier.

Programming software – Example of programming software

#To run the Code online, use
https://onecompiler.com/

An imperative procedural language supporting structured programming, lexical variable scope, and recursion with a static type system. It was designed to be compiled to provide low-level memory access, and language constructs that map efficiently to machine instructions, all with minimal runtime support.

Example 1: Hello World!

```
#include <stdio.h>

int main()
{
   printf("Hello World!");
   return 0;
}
```

Output:

```
Hello World!
```

Example 2: Find the averag

```
#include <stdio.h>
 // Function that return average of
given array.

double average(int a[], int n)
       {
       // Find the sum of array elemen
          int sum = 0;
          for (int i = 0; i < n; i++)
             sum += a[i];
          return (double)sum / n;
       }
```

```
int main()
{   // Input array
    int num[] = { 1, 2, 3, 4, 5 };

    // Size of array
    int n = sizeof(num) / sizeof(num[0]);

    // average(num, n) function is return
the average of the array.
    int avg = average(num, n);

    // Display average of given array
    printf("Average = %d ", avg);
    return 0;
}
```

Output:

```
Average = 3
```

System Software

Programming software – Example of programming software

#To run the Code online, use
https://onecompiler.com/

An imperative procedural language supporting structured programming, lexical variable scope, and recursion with a static type system. It was designed to be compiled to provide low-level memory access, and language constructs that map efficiently to machine instructions, all with minimal runtime support.

Example 1: Hello World!

```
#include <stdio.h>

int main()
{
   printf("Hello World!");
   return 0;
}
```

Output:

```
Hello World!
```

Example 2: Find the averag

```
#include <stdio.h>
 // Function that return average of
given array.

double average(int a[], int n)
       {
       // Find the sum of array elemen
          int sum = 0;
          for (int i = 0; i < n; i++)
             sum += a[i];
          return (double)sum / n;
       }
```

```
int main()
{   // Input array
    int num[] = { 1, 2, 3, 4, 5 };

    // Size of array
    int n = sizeof(num) / sizeof(num[0]);

    // average(num, n) function is return
the average of the array.
    int avg = average(num, n);

    // Display average of given array
    printf("Average = %d ", avg);
    return 0;
}
```

Output:

```
Average = 3
```

System Software

Programming software – Example of programming software

#To run the Code online, use
https://onecompiler.com/

C++ is an extension of the "C" programming language that is typically used for system/application software, game development, drivers, client-server applications and embedded firmware. C++ is complex and more difficult to learn. It remains in use in many legacy systems at large enterprises.

Example 1: Hello World!

```
#include <iostream>

using namespace std;

int main()

 {

  cout << "Hello World!";

  return 0;

}
```

Output:

```
Hello, World!
```

Example 2: Find the average

```
#include <iostream>
using namespace std;

// Function that return average of an
array.
double average(int a[], int n)
{
   // Find sum of array element
   int sum = 0;
   for (int i = 0; i < n; i++)
      sum += a[i];
   return (double)sum / n;
}
```

```
int main()
{
   int num[] = { 1, 2, 3, 4, 5};
   int n = sizeof(num) / sizeof(num[0]);

   cout << average(num, n) << endl;
   return 0;
}
```

Output:

```
3
```

System Software

Programming software – Example of programming software

#To run the Code online, use
https://onecompiler.com/

System Software

C# (pronounced "C sharp") is an object-oriented programming language from Microsoft designed to run on Microsoft's .NET platform to make development quicker and easier. C# is also used in video game development.

Example 1: Hello World!

```
using System;
namespace HelloWorld
{
  class Program
  {
    static void Main(string[] args)
    {
      Console.WriteLine("Hello World!");
    }
  }
}
```

Output:

```
Hello World!
```

Example 2: Find the average

```
using System;
using System.Linq;
using System.Collections.Generic;
class Purata {
  static void Main()
  {
    int[] numbers = { 1, 2, 3, 4, 5 };
                    int sum = 0;
            for (int i = 0; i < numbers.Length; i++)
                    {
                    sum += numbers[i];
                    }
            double average = (double)sum /
numbers.Length;
            Console.WriteLine("Average: " + average);
  }
}
```

Output:

```
Average:  3
```

Programming software – Example of programming software

#To run the Code online, use
https://onecompiler.com/

Python is a general-purpose programming language used for web development and as a support language for software developers. It's also widely used in scientific computing, data mining and machine learning. The continued growth and demand for machine learning developers may be driving the popularity of Python.

Example 1: Hello World!

```
print("Hello, World!")
```

Output:

```
Hello World!
```

Example 2: Find the average

```
numbers = [1, 2, 3, 4, 5]
average = sum(numbers)/len(numbers)
print("Average of list: ", average)
```

Output:

```
Average of list:  3.0
```

Programming software – Example of programming software

#To run the Code online, use
https://onecompiler.com/

PHP, a scripting language used on the server side, is used for web development to interact with MySQL databases.

Example 1: Hello World!

```
<?php
echo "Hello World!";
?>
```

Output:

```
Hello World!
```

Example 2: Find the average

```
<?php

$numbers = [1, 2, 3, 4, 5];
$average = array_sum($numbers) / count($numbers);
echo $average;

?>
```

Output:

3

Programming software – Example of programming software

#To run the Code online, use
https://onecompiler.com/

HTML5 is the latest version of Hypertext Markup Language, the code that controls the design of web pages. It allows for the combination of three kinds of code: HTML, which provides the structure of mobile and desktop web pages; Cascading Style Sheets (CSS) which determine how the pages look; and various implementations of JavaScript, which allows for custom programming inside web pages.

Example 1: Hello World!

```
<html>
<body>
<p>Hello World</p>
</body>
</html>
```

Hello World

Malicious software

Malicious software

- Malicious Software refers to any malicious program that causes harm to a computer system or network.
- Malicious Malware Software attacks a computer or network through viruses, worms, trojans, spyware, adware, or rootkits.
- These malicious programs steal, encrypt and delete sensitive data; alter or hijack core computing functions and monitor end users' computer activity.

Malicious software – Different Types Of Malicious Software

Sources:
https://www.techtarget.com/searchsecurity/definition/malware

Malicious software – Different Types Of Malicious Software

Computer Malicious

- A computer malicious is a software type that self-replicates and attaches itself to other files/programs. Malware Software can execute secretly when the host program/file is activated.

- The different types of Computer malicious are Memory-Resident viruses, Program File Viruses, Boot Sector Viruses, Stealth viruses, Macro viruses, and Email Viruses.

2. Worms

- A worm is a malicious software type similar to that of a computer malicious is a self-replicating program. However, in the case of worms, malicious automatically executes itself.

- Worms spread over a network and can launch a cumbersome and destructive attack within a short period.

Malicious software – Different Types Of Malicious Software

Trojan Horses

- Unlike a computer malicious or a worm – the trojan horse is a malicious software non-replicating program that appears legitimate.
- After gaining the trust, it secretly performs malicious and illicit activities when executed.
- Hackers use trojan horses to steal a user's password information and destroy data or programs on the hard disk. It is hard to detect!

Spyware/Adware

- Spyware is malicious software that secretly records user information and forwards it to third parties.
- The information gathered may cover files accessed on the computer, a user's online activities, or even the user's keystrokes.
- Adware is a malware software type as the name interprets displays advertising banners while a program is running. Adware can also work like spyware, it is deployed to gather confidential information. Basically, to spy on and gather information from a victim's computer.

Malicious software – Different Types Of Malicious Software

Rootkit

- A rootkit is a malicious software type that alters the regular functionality of an OS on a computer in a stealthy manner.
- The altering helps the hacker to take full control of the system and the hacker acts as the system administrator on the victim's system.
- Almost all the rootkits are designed to hide their existence.

Malicious software – How malware spreads?

Malware can get onto your computer in a number of different ways. Here are some common examples:

- Downloading free software from the Internet that secretly contains malware
- Downloading legitimate software that's secretly bundled with malware
- Visiting a website that's infected with malware
- Clicking a fake error message or pop-up window that starts a malware download
- Opening an email attachment that contains malware

There are many different ways that malware can spread, but that doesn't mean you're powerless to stop it.

Sources: Google Ads Help

Malicious software – Methods Of Protection Against Malicious Software

1) Keep your computer and software updated

1) Use a non-administrator account whenever possible

1) Think twice before clicking links or downloading anything

1) Be careful about opening email attachments or images

1) Don't trust pop-up windows that ask you to download software

1) Limit your file-sharing

1) Use antivirus software

Sources: Google Ads Help

Careers In IT

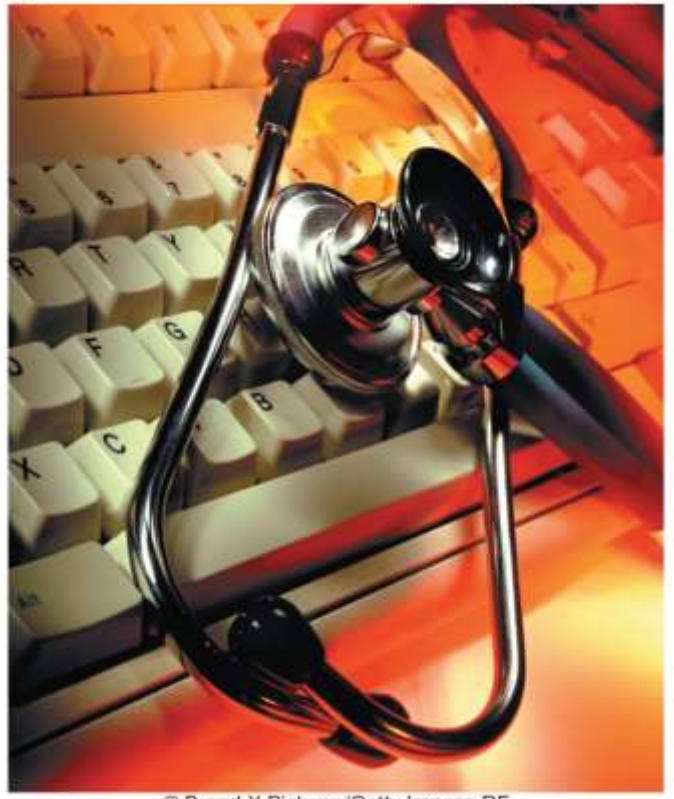
- Computer Support Specialist or Technical Support Specialist
  - Provide technical support to customers and other users
  - Resolve common networking problems and use troubleshooting programs to diagnose problems
  - Employers look for an advanced associate degree or bachelors degree, good analytical, customer service, communication and people skills
  - Computer support specialist can expect to earn $29K - $40K annually

Copyright © McGraw-Hill Education. Permission required for reproduction or display.
© Thomas Barwick/Getty Images RF

A Look to the Future – Self Healing Computers

- Self Healing Computers
  - Could mean an end to computer crashes and performance problems
  - Fix software problems
  - Reroute functions around broken hardware
- IBM's Automatic Computing Initiative (ACI)
  - Handles time-consuming maintenance
  - Self-regulating and virtually invisible
- Self-maintaining servers
  - Self-repairing
  - Self-updating
  - Self-protecting

Copyright © McGraw-Hill Education. Permission required for reproduction or display.



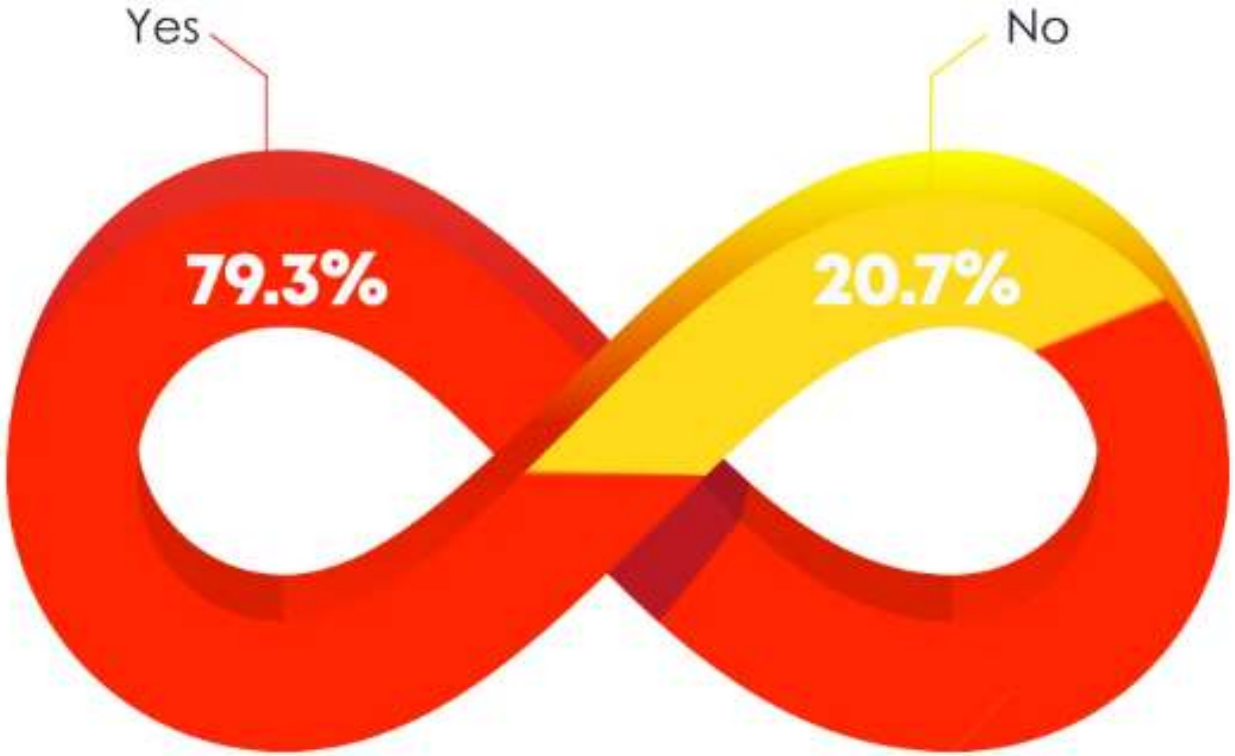
© Brand X Pictures/Getty Images RF

Trends in Software

- User Experience Design
  - User Experience Design (UXD or UED) is the process of enhancing user satisfaction of a software product by improving the usability, accessibility, and pleasure provided across each touch point.
- DevSecOps
  - DevSecOps (development, security, and operations) is an approach to software development (and culture) that embeds security throughout every stage of the DevOps pipeline.

Source: https://www.netsolutions.com/insights/the-future-of-software-development/

Trends in Software



Do You Apply DevSecOps?

Source: https://www.netsolutions.com/insights/the-future-of-software-development/

Trends in Software

- Omnichannel Experiences
  - Omnichannel experiences offer a seamless experience to the customer or user across various channels: retail, telephone, online, mobile, or social.
- The API Economy
  - The API economy is the use of APIs to achieve integration across services and data, allowing organizations to gain access to new services and products without building it themselves.
- Balanced Development Automation
  - Research from GitHub has demonstrated that automating software delivery alone can lead to a 31% faster merge and helps teams decrease time-to-market.

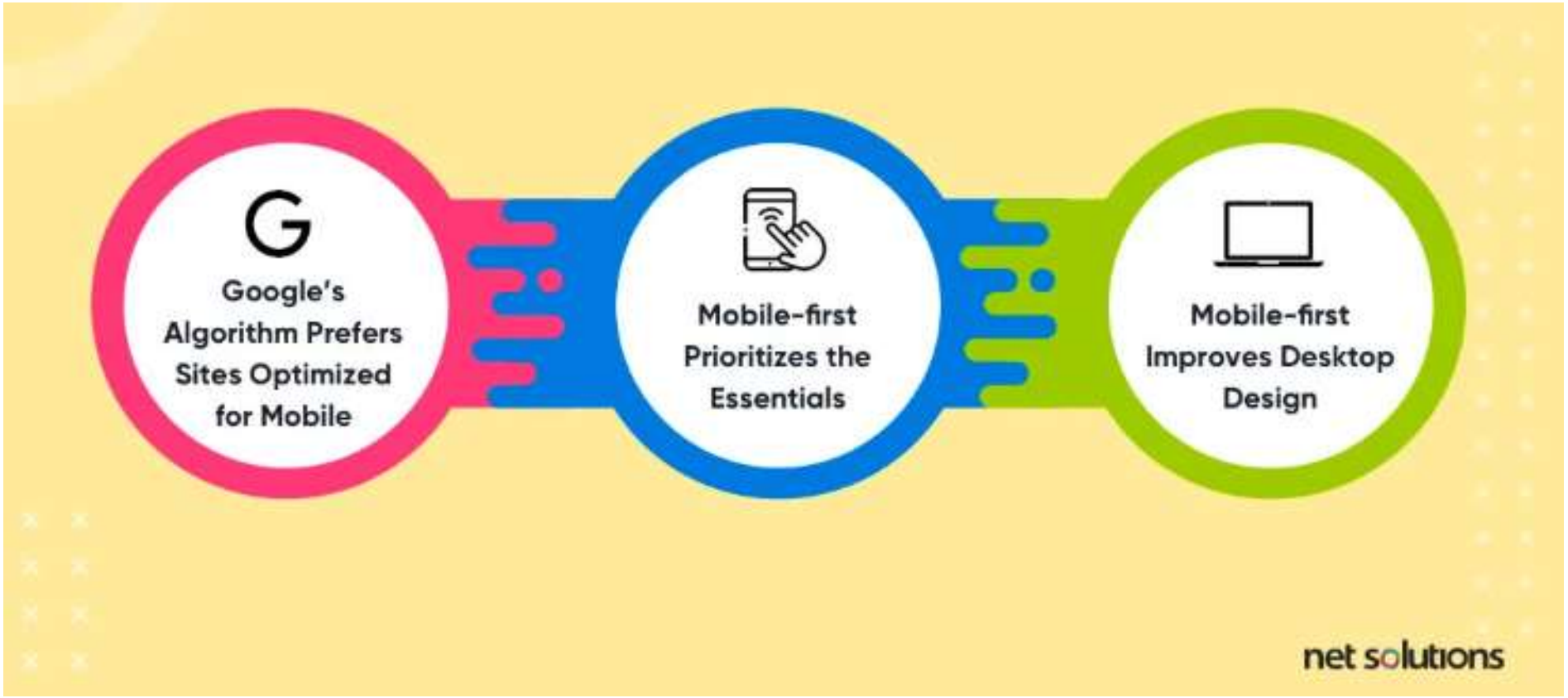
Source: https://www.netsolutions.com/insights/the-future-of-software-development/

Trends in Software

- Vulnerability Disclosure Programs
  - A vulnerability disclosure program (VDP) creates a secure channel for security issues and vulnerabilities to be reported following ISO standards around disclosure and handling and best practices around protecting whistleblowers.
- Mobile First
  - Mobile traffic now makes up 54.8% of all web traffic, so it's no longer about tagging a mobile website onto an existing design. Today, the approach needs to be mobile-first.

Source: https://www.netsolutions.com/insights/the-future-of-software-development/

Trends in Software



Mobile First Approach

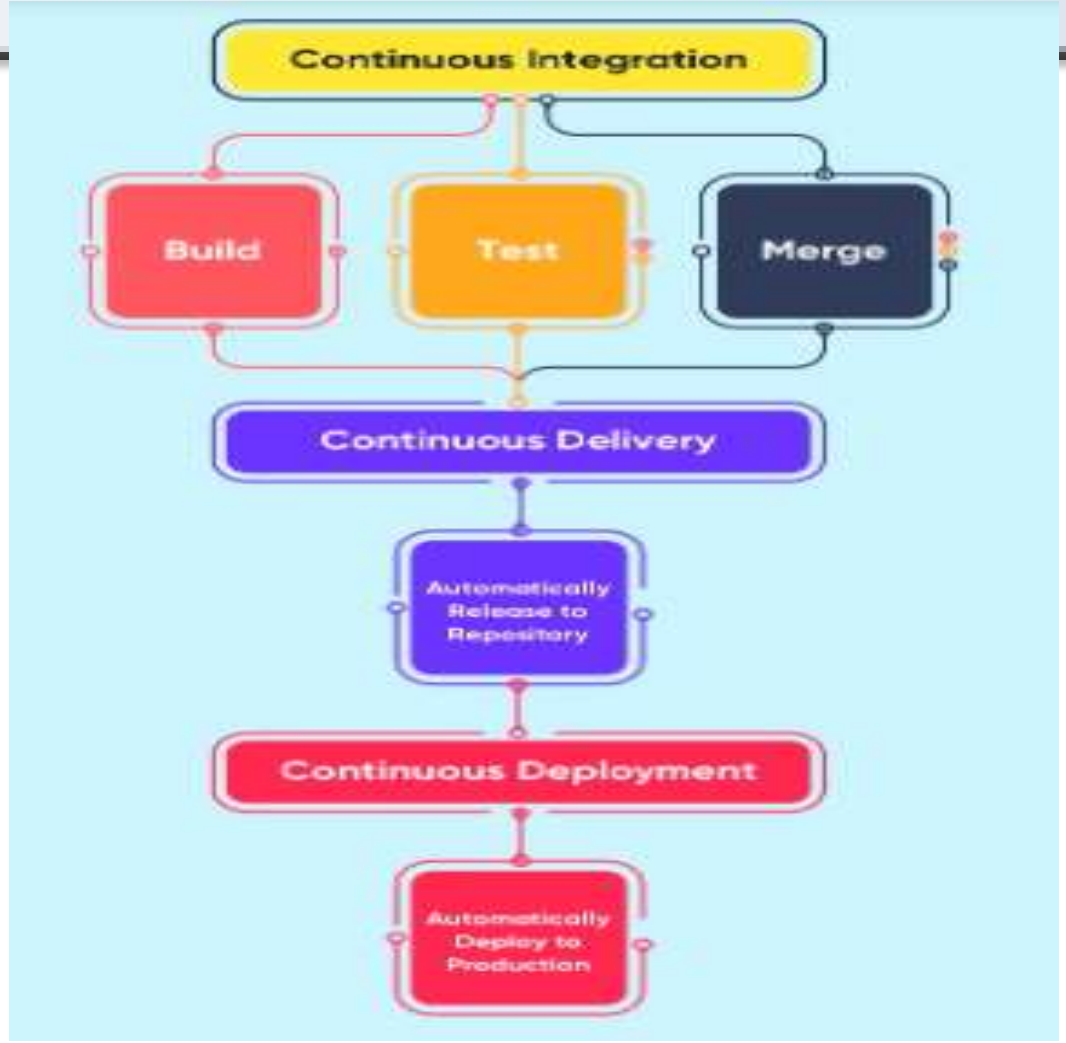
Source: https://www.netsolutions.com/insights/the-future-of-software-development/

Trends in Software

- Continuous Integration and Delivery
  - Continuous Integration, Continuous Delivery (CI/CD) is a way to introduce automation into the Agile and DevOps development process, a loop that implies constant updates to software to help improve quality and efficiency in the development process.
- Serverless Computing
  - Serverless computing leverages managed cloud infrastructure (AWS, Azure, Google App Engine) to build apps without having to host or maintain the servers.

Source: https://www.netsolutions.com/insights/the-future-of-software-development/

Trends in Software



CI/CD Development Process

Source: https://www.netsolutions.com/insights/the-future-of-software-development/

Trends in Software

- Blockchain
  - Blockchain, the use of a decentralized network of computers for digital transactions, is one of rapidly growing software development trends. In healthcare, blockchain could be one of the tools to help identify inaccuracies in patient data, reduce breaches, and help support patient access to health records.
- Expansion of the IoT
  - The Internet of Things (IoT) is expected to have a CAGR of 10.53% within the next 5 years, creating new platforms of "things" that can run software, but also changing how data is tracked and collected, how users engage across IoT devices (omnichannel expectations), and even impacting inventory management or shipping. All of this data must be translated into insight, with IoT spurring the need for AI and analytics.

Source: https://www.netsolutions.com/insights/the-future-of-software-development/

Trends in Software

- Escalation of the Edge Computing
  - Edge computing, the concept of a distributed IT architecture, places service provisioning, data, and intelligence closer to devices and users to help boost performance.
- Quantum Computing
  - Quantum computing leverages quantum mechanics (superposition, interference, entanglement) to perform calculations, with the number of organizations and governments investing in quantum computing starting to improve processes and operations.

Source: https://www.netsolutions.com/insights/the-future-of-software-development/

Trends in Software

- Big Data
  - Big data is a field that works with large or complex data sets of structured and unstructured data using advanced techniques. Look toward concepts such as the data fabric (the flexible integration of data sources), understanding the options of the data marketplace, analytics, edge infrastructure, and AI.
- Augmented Reality (AR)
  - Virtual Reality (VR) and Augmented Reality (AR) continue to find new avenues for growth, through gaming applications but also other use cases including eCommerce, healthcare, manufacturing, and education. Apple is also rumoured to be releasing an AR/VR headset in 2022.

Source: https://www.netsolutions.com/insights/the-future-of-software-development/

Trends in Software

- Low-Code / No-Code Platforms
  - Low-code development platforms (LCDP) and no-code development platforms (NCDP) offer pre-built blocks that can be dragged and dropped (visual environment) to assist in rapid development of mobile and web apps both by professional developers (for speed) and by those outside the IT department.

Source: https://www.netsolutions.com/insights/the-future-of-software-development/

Trends in Software

- Artificial Intelligence (AI)
  - AI is the basis for mimicking human intelligence processes by creating and applying algorithms built into a dynamic computing environment.
  - AI is trying to make computers think and act like humans.
    - Achieving this end requires three key components:
    - Computational systems
    - Data and data management
    - Advanced AI algorithms (code)
  - The more humanlike the desired outcome, the more data and processing power required

Source: https://www.netapp.com/artificial-intelligence/what-is-artificial-intelligence/

Trends in Software

Machine Learning: An approach to AI in which a computer algorithm (a set of rules and procedures) is developed to analyse and make predictions from data fed into the system.

- Machine learning-based technologies are routinely used every day, such as personalised news feeds and traffic prediction maps.

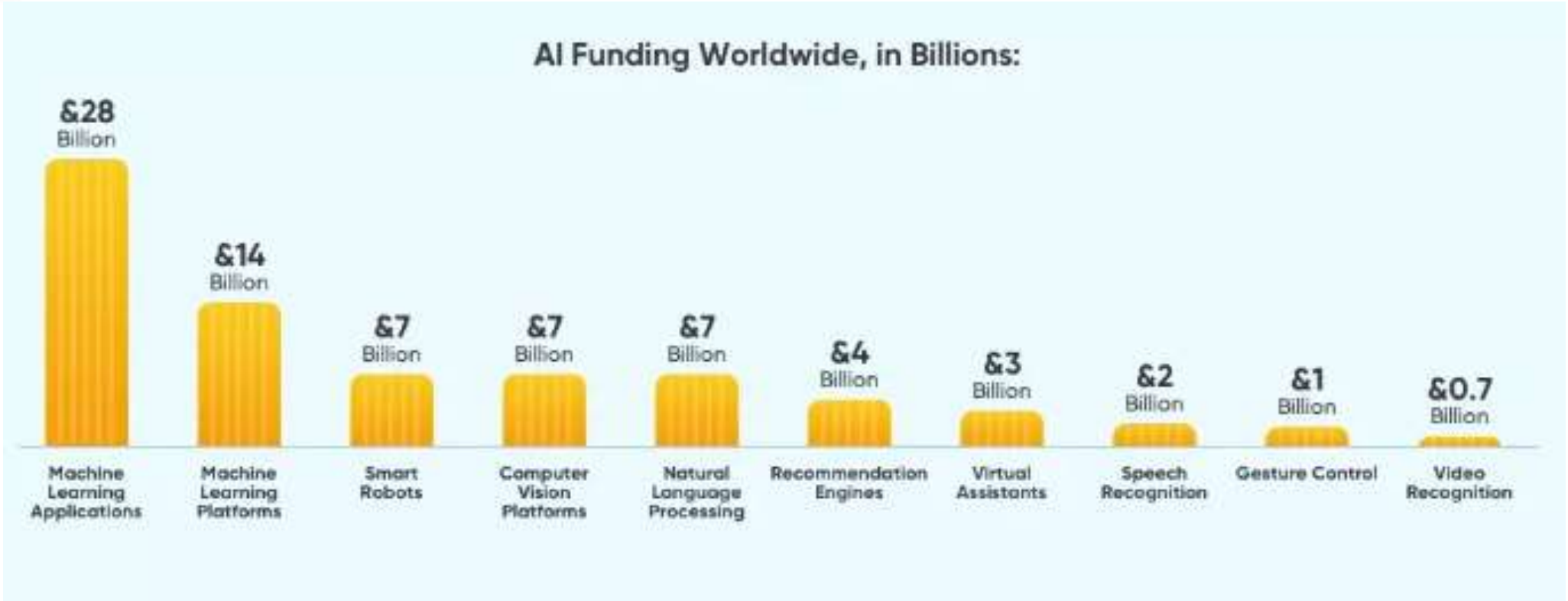
Neural Networks: A machine learning approach modelled after the brain in which algorithms process signals via interconnected nodes called artificial neurons.

- Artificial neural networks have been used successfully to mimic biological nervous systems to recognise and predict patterns of neural signals involved in brain function.

Deep Learning: A form of machine learning that uses many layers of computation to form what is described as a deep neural network, capable of learning from large amounts of complex, unstructured data.

- Deep neural networks are responsible for voice-controlled virtual assistants and self-driving vehicles, which learn to recognise traffic signs.

Trends in Software



Machine Learning Tops AI Funding Worldwide

Source: https://www.netsolutions.com/insights/the-future-of-software-development/

Trends in Software

Top Machine Learning Use Cases

Source: https://www.netsolutions.com/insights/the-future-of-software-development/

Trends in Software

EXAMPLE OF AI

1. ChatGPT
   - is a natural language processing tool driven by AI technology that allows you to have human-like conversations and much more with the chatbot.

   - The language model can answer questions and assist you with tasks, such as composing emails, essays, and code.

How to start using ChatGPT

https://www.youtube.com/watch?v=jHv63Uvk5VA

Open-Ended Questions (Page 1 of 2)

1. Describe system software. Discuss each of the four types of system programs.

1. Define operating systems. Describe the basic features and the three categories of operating systems.

1. What are mobile operating systems? Describe leading mobile operating systems.

Open-Ended Questions (Page 2 of 2)

4. What are desktop operating systems? Compare Windows, Mac OS, Linux and Chrome OS. Discuss virtualization.

4. Discuss utilities. What are the most essential utilities? What is a utility suite?