

Software Development Life Cycle

Lecture 1: Software and Software Engineering

DR. YOUNG SAENG PARK
2021/2022



About Software



Programming Languages



Software Engineering



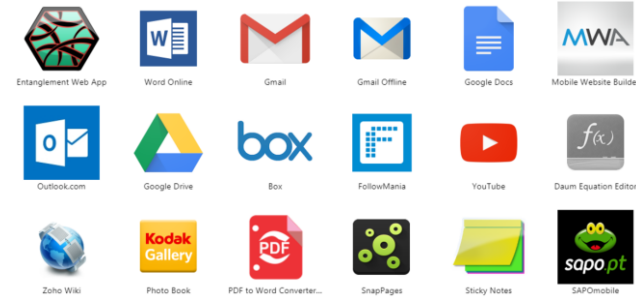
Software Engineer vs Developer

Software and Software Engineering

About Software

Software and Application

- ☐ **Program** is a **set of instructions** telling a computer **how to work**.
- ☐ **Application** is a program or group of programs **performing a specific task for end-users**.
- ☐ **Software** is a collection of programs coordinating with the hardware to **run the machine for any purpose**.
- ☐ Software is not often **operating system based** but application is operating systems based.
- ☐ Software may be **executable** or may not be executable but application is always executable.
- ☐ Software may not need **user interaction** but application need user interaction for functioning.
- ☐ **All the software are not applications** but **all the application are software**.



Source: TurboFuture

Types of Software (1)

Stand-alone software

- **Run on a personal computer** or run on a **mobile device**.
- Include all **necessary functionalities**.
- **Not always need to be connected** to a network.
- e.g.) any office applications, image and text editors, web browsers, etc

Interactive transaction-based software

- **Execute on a remote computer or server** via the Internet and **access the resources** from their own computers.
- Often **incorporate a large data storage** that is accessed and updated in each transaction.
- e.g.) **web applications** such as e-commerce application to buy goods and services, **business systems** such as enterprise resource planning, **mail and data sharing systems** through a web browser or special-purpose client typically based on cloud-based services.



Source: Fiverr

Types of Software (2)

Embedded control software

- Control and manage **hardware devices**.
- Usually has **time and memory constraints** because of the limited computing capabilities.
- **May not use operating system** unlike other software.
- e.g.) software controlling antilock braking in a car, software in a microwave oven, etc.



Source: SSLA

Batch processing software

- Designed to **process data in large batches**.
- Can **be scheduled to run** as resources permit.
- Typically **process large number of individual inputs** to create corresponding outputs.
- e.g.) periodic billing systems such as phone billing system and water billing systems.

Types of Software (3)

☐ Entertainment software

- Use to **entertain the user**, providing a form of **amusement**.
- Run on a **computer** or **special-purpose control hardware**.
- Has attention to **VR** (Virtual Reality), **AR** (Augmented Reality), **MR** (Mixed Reality) and **XR** (Extended Reality).
- e.g.) video games, mobile game, digital pets, digital media applications, etc.



Source: Roblox

☐ Software for modelling and simulation

- Create and analyse a **digital prototype** of a physical model to **predict** its performance in the real world.
- Typically require computationally **high-performance parallel system for execution**.
- e.g.) climate and weather simulation, finances and investment market modelling, virtual manufacturing simulation, etc.



Source: Inverse

Types of Software (4)

Data collection software

- **Collect qualitative and quantitative data** in an electronic form and also send the data to other systems for processing.
- May **interact with sensors** and often is installed on **edge devices** in a remote location.
- Allow data to be quickly exported for **data analysis and reporting**.
- e.g.) website visits, weather data collection, performance measurement systems, etc.



Source: Sprintally

Systems of systems

- Is a **collection of systems**, each capable of independent operations.
- Interoperate together to **achieve additional desired capabilities**.
- Is often very **large-scale distributed systems**.
- e.g.) enterprise resource planning, air traffic control, etc



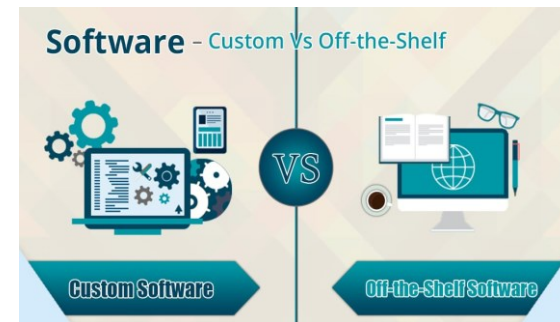
Generic Products

- Software that is **sold to any customer** who wishes to buy and use it.
- Examples – PC software such as graphics programs, project management tools, CAD software, MS offices, etc.
- **The specification** of what the software should do is **owned by the software developer** and **decisions** on software change are **made by the developer**.



Customised Products

- Software that is **commissioned by a specific customer** to meet their own needs.
- Examples – embedded control systems, air traffic control software, traffic monitoring systems, etc.
- **The specification** of what the software should do is **owned by the customer** for the software and **they make decision** on software changes that are required.



Source: Viftech



Software Failures

- ❑ **Software failure** occurs when software does **not provide the expected result** with respect to specification input values.
- ❑ Depending on the failure impact to systems, the levels of failures are **catastrophic, critical, major** or **minor**.



Source: dreamstime

- ❑ Typical reasons for software failures are:
 - Poor to use software engineering methods
 - Unrealistic software goals
 - Poor to handle the project's complexity
 - Poor estimates of resources
 - Use of immature technology
 - Badly defined system requirements
 - Poor reporting the project's status
 - Poor communication among customers, developers and users
 - Poor management of risks
 - Commercial pressures

....



Good Software

☐ Maintainability

- Software should be written in such a way **to meet the changing needs from customers** because software change is inevitable in business environment.

☐ Dependability

- Dependable software, which has **reliability, security and safety**, should **not cause economic damage in the event of system failure**.

☐ Efficiency

- Software should **not waste system resources** such as memory and processor cycle. Efficiency includes **responsiveness, processing time, memory utilisation**, etc.

☐ Acceptability

- Software should **be acceptable to the type of uses**. It must be **understandable, usable and compatible** with other systems.



Source: DesignRush

Programming Language

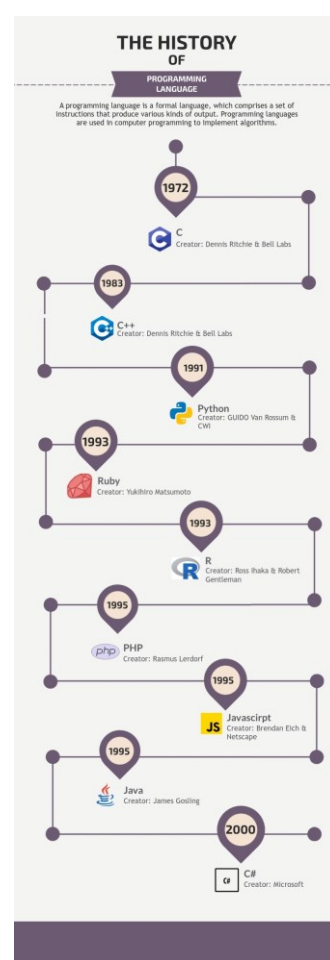
- ❑ Software is developed using **one or more programming language** to address business or personal objectives.
- ❑ A programming language is **formal language comprising a set of instructions** that produce various kind of machine code output via **compiler** or **interpreter**.
- ❑ **Thousands of different programming** languages are available for software development and more are being created every year.
- ❑ Many programming languages are written in the **imperative** form (sequence of operations to perform) or in the **declarative** form (the desired result is specified).
- ❑ Programming languages fall into **two different classifications**: **Low-level** (closer to machine code - difficult for humans to read but fast) and **High-level** (closer to how human communicate – easier to program but slow).



Source: TechGig

Programming Language Generations

- ☐ **First Generation (1GL)** is machine-level programming language (e.g. binary machine code).
- ☐ **Second Generation (2GL)** is a low-level programming language (e.g. assembly language).
- ☐ **Third Generation (3GL)** is a high-level computer programming language which tends to be **much more machine independent and more programmer friendly** (e.g., C, C++, Java, Python, PHP, Perl, C#, BASIC, Fortran, Algol, Cobol, etc.).
- ☐ **Fourth Generation (4GL)** is any programming language that tends to be **specialised toward very specific programming domains** such as report generation, GUI development or web development (e.g. SQL, PowerBuilder, Oracle Report, PL/SQL, etc.).
- ☐ **Fifth Generation (5GL)** is any programming language **based on problem-solving using constraints** given to the program rather than using an algorithm written by a programmer (e.g. OPS5, Mercury, etc.).



Source: Kantar Information

Programming Language Languages (1)

Python

- It is an interpreted high-level general-purpose programming language. It can be used in a variety of fields **from data science and machine learning to web development** and is a great first language to learn.

Java

- It is a high-level, class-based, **object-oriented programming language** that is designed to have as few implementation dependencies as possible. It can be used for many things including **mobile application, software development, web development** and **large systems development**.

JavaScript

- It is a **front-end** and **back-end** friendly language that **enables web application**, game development and mobile applications. JavaScript is **one of the core technologies of the World Wide Web** along side HTML and CSS.

Worldwide, Mar 2021 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Python	30.17 %	-0.2 %
2		Java	17.18 %	-1.2 %
3		JavaScript	8.21 %	+0.2 %
4		C#	6.76 %	-0.6 %
5	↑	C/C++	6.71 %	+0.8 %
6	↓	PHP	6.13 %	+0.0 %
7		R	3.81 %	+0.0 %
8		Objective-C	3.56 %	+1.1 %
9		Swift	1.82 %	-0.4 %
10	↑	Matlab	1.8 %	-0.0 %
11	↑	Kotlin	1.76 %	+0.2 %
12	↓↓	TypeScript	1.74 %	-0.1 %
13	↑	Go	1.34 %	+0.0 %
14	↓	VBA	1.22 %	-0.1 %
15		Ruby	1.13 %	-0.1 %

Source: Popularity of Programming Language

Programming Language Languages (2)



C#

- It is a general-purpose, **multi-paradigm programming language** developed around 200 **by Microsoft as part of its .NET**. It can be used for a wide variety of application, including game development, enterprise software, video games, mobile apps, etc.



C++

- It a general-purpose programming language **as an extension of the C programming language**. It is used in a wide range of industries, including VR, software and game development, robotics, and **scientific computing due to fast and low memory usage**.



PHP

- It is a general-purpose scripting language especially **suited to web development**. It is used to manage dynamic content, databases, session tracking, major protocols such as POP3, IMAP and LDAP, and even build entire **e-commerce sites**.

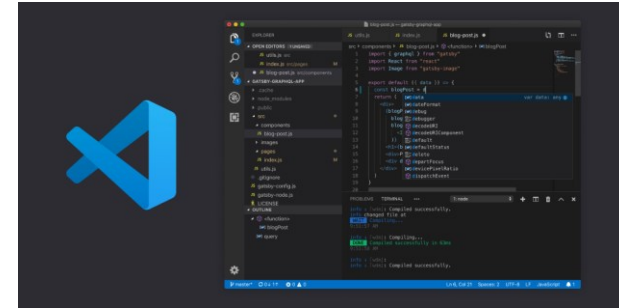
Worldwide, Mar 2021 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Python	30.17 %	-0.2 %
2		Java	17.18 %	-1.2 %
3		JavaScript	8.21 %	+0.2 %
4		C#	6.76 %	-0.6 %
5	↑	C/C++	6.71 %	+0.8 %
6	↓	PHP	6.13 %	+0.0 %
7		R	3.81 %	+0.0 %
8		Objective-C	3.56 %	+1.1 %
9		Swift	1.82 %	-0.4 %
10	↑	Matlab	1.8 %	-0.0 %
11	↑	Kotlin	1.76 %	+0.2 %
12	↓↓	TypeScript	1.74 %	-0.1 %
13	↑	Go	1.34 %	+0.0 %
14	↓	VBA	1.22 %	-0.1 %
15		Ruby	1.13 %	-0.1 %

Source: *Popularity of Programming Language*

Software Development Tools

- ☐ **Software development tools** are used by the software developers for creating, editing, maintaining, supporting and debugging applications, frameworks and programs.
- ☐ The software development tools can be of many forms such as **linkers, compilers, code editors, GUI designer, assemblers, debugger, performance analysis tools**, etc.
- ☐ The tools may be combined within a framework called an **Interactive Development Environment** or **IDE**. This provides a **common set of facilities** that tools can use so that it is easier to communicate and operate in an integrated way.
- ☐ Using the development tools, a developer can **easily maintain the workflow** of the project.
- ☐ Using the tools in software development process, the outcome of the projects will be **more productive**.

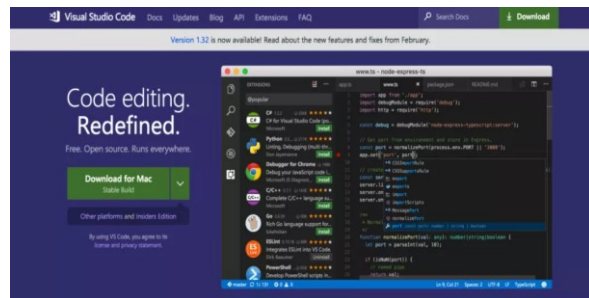
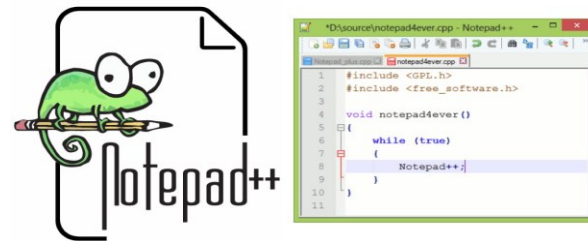
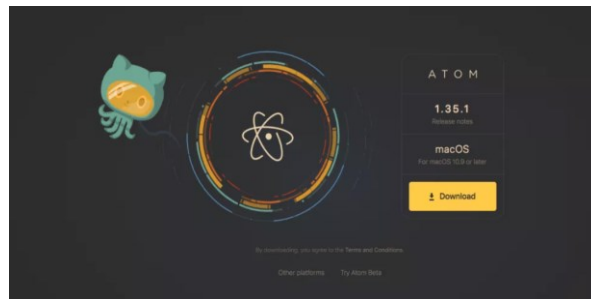
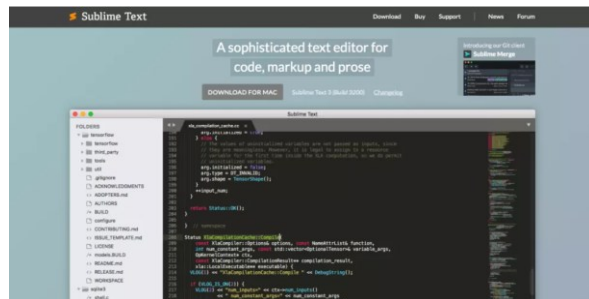


Source: Visual Studio Code



Source: Microsoft

Popular Code Editors



Software and Software Engineering

Programming Languages

Programming Language – Python

- ☐ **Python** is a simple, general purpose, high level, and object-oriented programming language.
- ☐ Python is an **interpreted scripting language**, invented by *Guido Van Rossum* as the founder of Python programming.
- ☐ The name of Python came from BBC comedy show, “**Monty Python’s Flying Circus**”.
- ☐ Python is **easy to learn powerful and versatile scripting language**, which makes it attractive for application development.
- ☐ Python’s syntax and dynamic typing with **its interpreted nature** make it an ideal language for **scripting and rapid application development**.
- ☐ Python **supports multiple programming pattern**, including object-oriented, functional and procedural programming styles.



```
def func():  
    statement 1  
    statement 2  
    .....  
    .....  
    statement N
```

```
print("Hello World")
```

Programming Language – Python Features (1)

☐ Easy to learn and use

- Its syntax is straightforward and much the same as the English language. There is **no use of the semicolon or curly-bracket**.

☐ Interpreted language

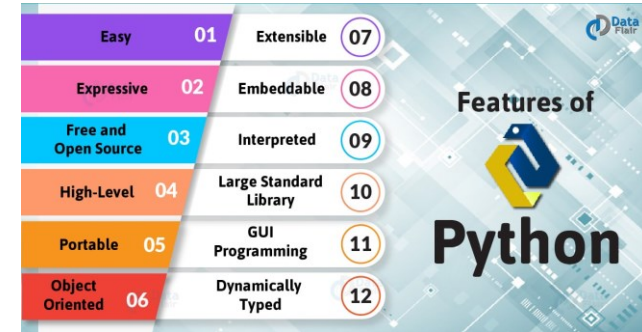
- Python program is executed one line at a time. This will help to **clear error easily**.

☐ Cross platform language

- Python can **run equally on different platforms** such as Windows, Linux, and Mac. So, Python is a portable language.

☐ Free and open source

- Python is **freely available for everyone**. It has a **large community** across the world that is dedicatedly working towards make new python modules and functions.



Source: DataFlair

Programming Language – Python Features (2)

☐ Object-oriented language

- Python **supports object-oriented language**. So, it supports inheritance, polymorphism, encapsulation, etc.

☐ Dynamic memory allocation

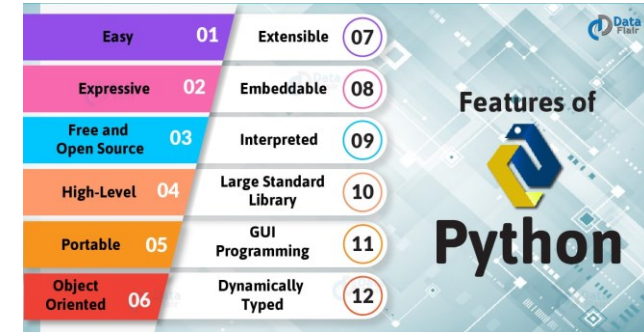
- There is **no need to specify the data-type** of the variable. When it assigns some value to the variable, it automatically allocates the memory to the variable at run time.

☐ Large standard library

- It provides a **vast range of libraries** for the various fields such as machine learning, web programming, etc.

☐ GUI programming support

- Graphical user interface is use** for the developing desktop application.

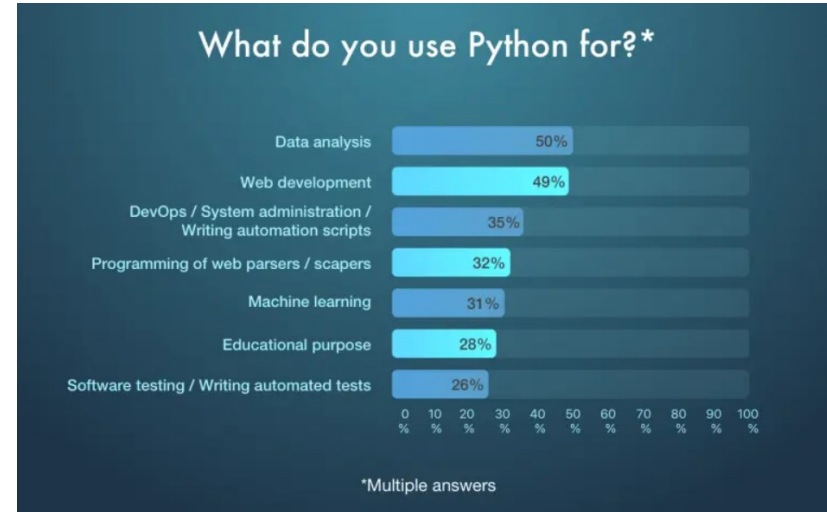


Source: DataFlair

Programming Language – Python Applications

☐ Python is used in almost every technical field:

- Data Science
- Data Mining
- Desktop Applications
- Console-based Applications
- Mobile Applications
- Web Applications,
- Enterprise Applications.
- Artificial Intelligence
- Machine Learning
- Computer Vision or Image Processing Applications.
- 3D CAD Applications
- ...

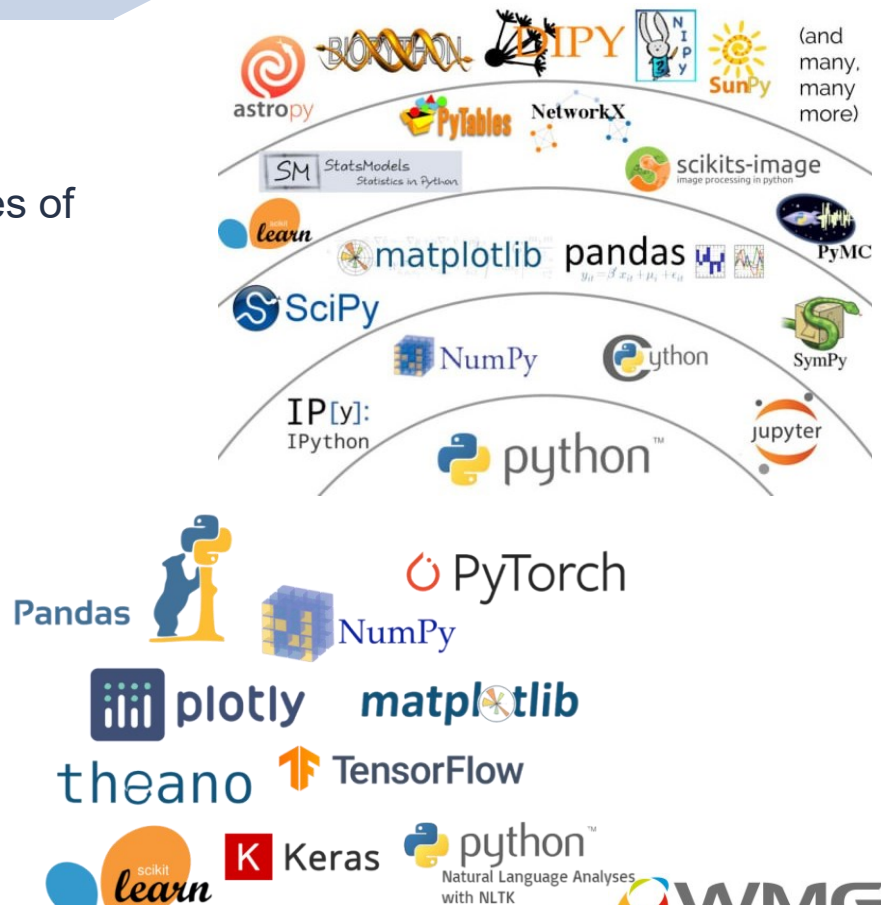


Source: BoTree Technologies



Programming Language – Python Libraries and Frameworks

- Python has **wide range of libraries and frameworks** widely used in various fields.
- There are some popular frameworks and libraries of Python:
 - Web development (Server-side)**
 - ✓ Django, Flask, Pyramid, CherryPy, etc.
 - GUIs based applications**
 - ✓ Tk, PyGTK, PyQt, PyJs, etc.
 - Machine Learning**
 - ✓ TensorFlow, PyTorch, Scikit-learn, Matplotlib, Keras, etc.
 - Mathematics & Data Analysis**
 - ✓ NumPy, SciPy, Pandas, etc.



Programming Language – Java

- ☐ Java is a **general-purpose programming** language intended to let application developers “**write once, run anywhere**”.
- ☐ It means that **compiled Java code can run on all platforms** that support Java **without the need for recompilation**.
- ☐ Java applications are typically **compiled to bytecode** that can **run on any Java Virtual Machine (JVM)** regardless of the underlying computer architecture.
- ☐ End users commonly use a **Java Runtime Environment (JRE)** as Java Virtual Machine installed on their machine for standalone Java applications.
- ☐ The syntax of Java is **similar to C and C++** but has fewer low-level facilities than either of them.
- ☐ Java developed by Sun Microsystems in 1995 is now **the subsidiary of Oracle**.



```
class Simple{  
    public static void main(String args[]){  
        System.out.println("Hello Java");  
    }  
}
```


❑ There are mainly 4 types of applications that can be created using Java programming:

- **Standalone Application**

- ✓ These are traditional software installed on every machine. **AWT and Swing** are used in Java for creating standalone applications.

- **Web Application**

- ✓ An application running on the server side creates a dynamic web page. Currently, **Java Server Page (JSP)**, Struts, Spring, Hibernate, JSF, etc. technologies are used for creating web applications in Java

- **Enterprise Application**

- ✓ An application that is distributed in nature such as banking applications, etc. is called enterprise application. It has advantages like high-level security, load balancing, and clustering. In Java, **Java Enterprise Bean (EJB)** is used for creating enterprise applications.

- **Mobile Application**

- ✓ **Android** and Java ME are used for creating mobile applications.

Programming Language – Java Features (1)

☐ Simple for C++ programmer

- Java syntax is **based on C++**. It is easier for programmers for C++ programmers. There is no need to remove unreferenced objects because of an **automatic garbage collection** in Java.

☐ Object-oriented

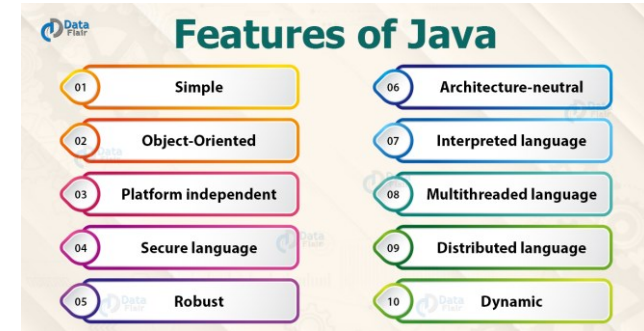
- Java is an **object-oriented programming** language. Everything in Java is an object.

☐ Platform independent

- Java code can be **executed on multiple platform using JVM** and is compiled by the compiler and converted into bytecode which is a platform independent.

☐ Secured

- Java language provides **security capabilities** such as Class Loader, Bytecode Verifier, and Security Manager **by default via JVM**.



Source: DataFlair

Programming Language – Java Features (2)

☐ Robust

- It uses **strong memory management** and there are **exception handling** and **type checking** mechanism in Java.

☐ Portable

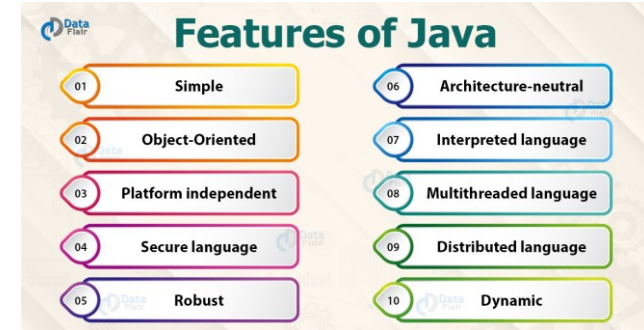
- Java is portable because it facilitates to **carry the Java bytecode to any platform**. It doesn't require any implementation.

☐ High performance

- Java is faster than other traditional interpreted programming languages because **Java bytecode is close to native code**. It is still a little bit slower than a compiled language like C++.

☐ Dynamic

- It supports the **dynamic loading of classes**, meaning that classes are loaded on demand. It also **supports other languages** such as C and C++ using **native** keyword.



Source: DataFlair

Programming Language – JavaScript

- ☐ **JavaScript** is a powerful and flexible programming language which is **used by websites**.
- ☐ Alongside HTML and CSS, JavaScript is **one of the core technologies of the World Wide Web**.
- ☐ It is interpreted programming language that **enables dynamic interactivity on websites** when applied to an **HTML document**.
- ☐ Users can build **modern web applications to interact** directly without reloading the page every time.
- ☐ All major web browsers such as Chrome, Microsoft Edge, Firefox have a **dedicated JavaScript Engine to execute the code** the user's device.
- ☐ JavaScript engine were **originally used in web browsers**, but they are **now core components of other software systems**, notably **servers** and a **variety of applications** using **Node JS**.



```
<script>  
document.write("Hello JavaScript by JavaScript");  
</script>
```

Programming Language – JavaScript Features (1)

Light weight scripting language

- JavaScript is a **lightweight scripting language** because it is made for **data handling at the browser only**.

Dynamic typing

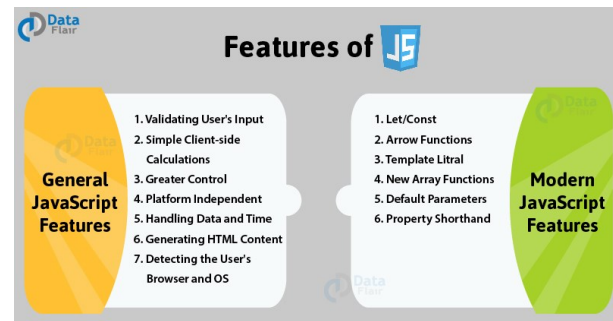
- JavaScript **supports dynamic typing** which means types of the variable are defined based on the stored value.

Object oriented programming support

- The latest JavaScript (string from ES6) **includes the concept of class and OOP**. However, JavaScript developers **rarely use this feature** rather than prototype based style.

Functional style

- JavaScript **uses a functional approach**. Functions in JavaScript can **be used as objects** and can **be passed to other functions** too.



Source: DataFlair

Programming Language – JavaScript Features (2)

Prototype-based language

- JavaScript is a **prototype-based scripting language**. So, JavaScript uses prototypes instead of classes or inheritance.

Async processing

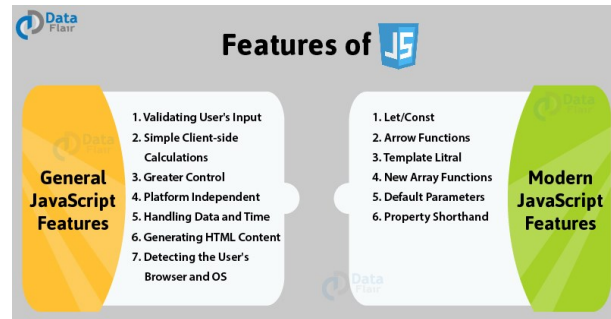
- JavaScript supports **Promise which enables asynchronous requests** wherein a request is initiated and JavaScript **does not have to wait for the response**.

Client-side validations

- JavaScript is **used for implementing client-side validation** to make sure that users enter the correct value. This feature is still widely used because every website need to check values.

Interpreted language

- JavaScript is an interpreted language, meaning **the script written inside JavaScript is processed line by line** using a built-in component of the Web browser.



Source: DataFlair

☐ Node JS

- Node JS is powerful JavaScript framework which **allows to develop a server-side component using JavaScript**. The biggest advantage of Node JS is that a programmer **can develop a web application end-to-end using the same language**.

☐ React JS

- React is **Facebook's component-based web development framework** for making UIs and offering declarative views, which makes the code more predictable and easier to debug.

☐ Vue JS

- Vue JS is another open-source progressive JavaScript framework for **building interactive user interfaces**, similar to React.

☐ jQuery

- jQuery is probably **the most popular JavaScript library** out there which provides many features for modern-day development.



Source: devRant

☐ Angular

- Angular previously known as Angular JS is a **single web development framework developed by Google** for both **desktop and mobile web applications**.

☐ Ember JS

- Ember JS is popular, open-source JavaScript web framework which is **based on the Model and View pattern**.

☐ D3 JS

- D3 JS is a JavaScript **library for producing dynamic, interactive data visualisation in web browsers** as D3 allows to bind data to Document Object Model (DOM) and then apply data-driven transformations to the document.

☐ Express JS

- Express JS is a **back end web application framework for Node JS**. It is used for easier creation of web applications and services.



Source: devRant

Programming Language – C#

- ❑ **C#** is a general-purpose, modern and **object-oriented programming language developed by Microsoft**.
- ❑ It is **based on C++ and Java**, but it has many **additional extensions** used to perform component oriented programming approach.
- ❑ C# programs **run on .NET**, a virtual execution system called the **common language runtime (CLR)** and a set of class libraries.
- ❑ **The CLR** is the implementation by Microsoft of the **common language infrastructure (CLI)**, an international standard.
- ❑ A programmer can **develop different types of secured and robust applications**: Windows, Web, Distributed, Web service and Game applications, etc.
- ❑ C# can create **robust and durable applications** using garbage collection, exception handling, lambda expression, language integrated query expression, asynchronous operations, etc.



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

Programming Language – C# Features (1)

Simple

- C# is a simple language in the sense that it provides **structured** approach, **rich set of library functions**, **data types**, etc.

Modern programming language

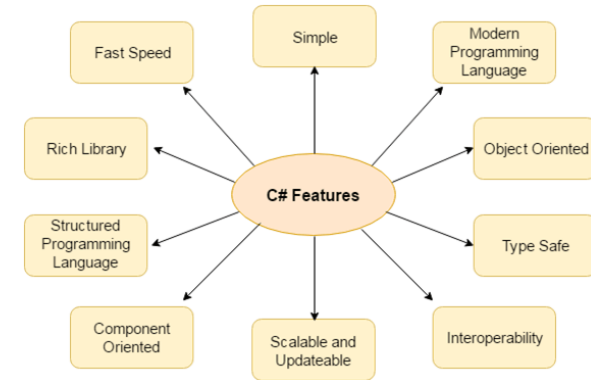
- C# programming is based upon the current trend and it is very **powerful and simple for building scalable, interoperable and robust applications**.

Object oriented

- C# is **object oriented programming language**. OOPs makes development and maintenance easier where as in procedure oriented programming it is not easy to manage if code grows as project size grow.

Type safe

- C# type safe code can **only access the memory location**. Therefore, it improves a security of the program.



Source: JavaTpoint

Programming Language – C# Features (2)

Component oriented

- C# is component oriented programming language. It is the predominant software development **methodology used to develop more robust and highly scalable applications.**

Rich library

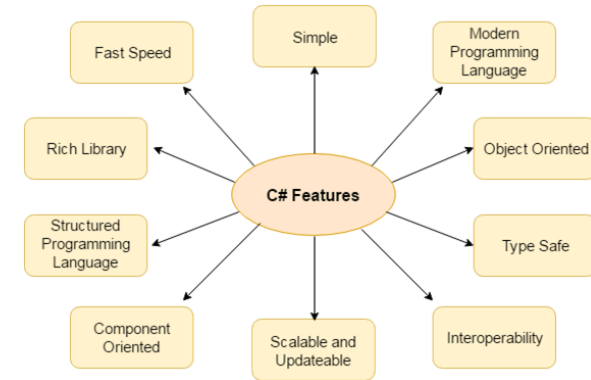
- C# provides **a lot of in-built functions** that makes the development fast.

Integrated development environment

- **Microsoft provides integrated development tools like Visual Studio, Visual Studio Code.** Using these tools, a programmer can write all kinds of C# programs from simple command-line applications to more complex applications.

Interoperability

- **Component from VB NET and other managed code languages and directly be used in C#.**



Source: JavaTpoint

Programming Language – C++

- ☐ **C++** is a general purpose programming language as an **extension of the C programming** language.
- ☐ C++ programming is a superset of C, meaning **any valid C program is also a valid C++ program**.
- ☐ C++ was designed with an orientation toward **system programming, embedded software** and **large system** with performance, efficiency and flexibility.
- ☐ C++ is a **middle-level language**, as it encapsulates both high and low level language features.
- ☐ Modern C++ supports **object-oriented, generic, and functional features**.
- ☐ C++ is **useful in resource-constrained applications** such as video games, servers, and performance-critical applications.



Source: Spacemacs

```
#include <iostream.h>
#include <conio.h>

void main() {
    clrscr();
    cout << "Welcome to C++ Programming.";
    getch();
}
```

Programming Language – C++ Features (1)

Object Oriented

- C++ is an **object oriented language**, unlike C which is a procedural language. It helps data abstraction, data encapsulation, data hiding, and polymorphism.

Compiler based

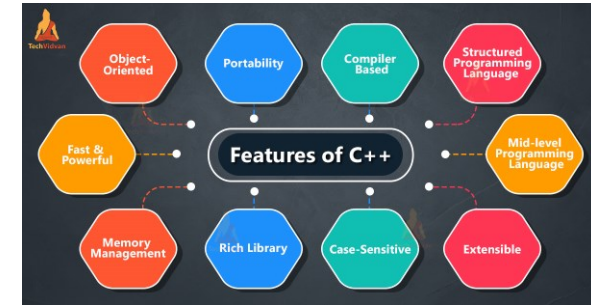
- Unlike Java and Python that are interpreter based, C++ is a **compiler based language** and hence it is **relatively much faster than Python and Java**.

Memory management

- It supports the feature of **dynamic memory allocation**. In C++ language, it allows to **free the allocated memory at any time** by calling the `free()` function.

Speed

- C++ is compiler based hence it is **much faster than other programming language**.



Source: TechVidvan

Programming Language – C++ Features (2)

☐ Easy to learn for C developer

- It might be **easy to learn C++ for C developer** as C++ includes C syntax and function libraries.

☐ Mid-level programming

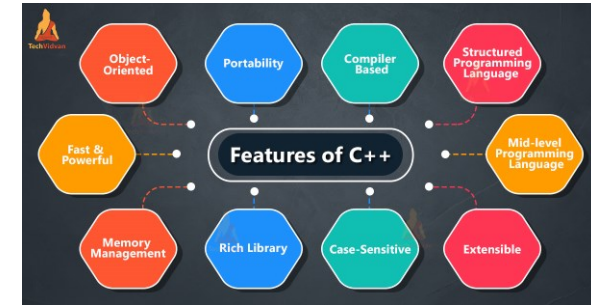
- C++ is used to **do low level programming**. It is used to develop **system applications** such as kernel, driver, etc. It also supports the feature of high level language.

☐ Base language for others

- C++ can be **the base language for many other programming languages** that supports the feature of object oriented programming.

☐ Existence of libraries

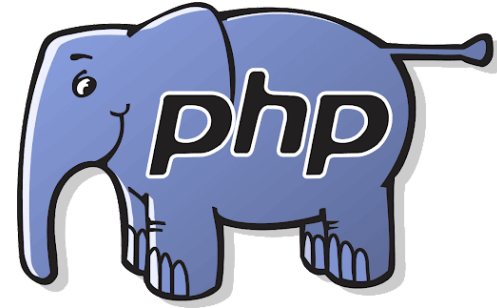
- The C++ programming languages **offers a library full of in-built functions** that make programs easy for the programmer.



Source: TechVidvan

Programming Language – PHP

- ☐ **PHP** is a server side scripting language which is used to **develop the dynamic web applications**.
- ☐ PHP is one of the **most widely used server side scripting language** for web development and popular websites are like Facebook, Yahoo, Wikipedia, etc.
- ☐ PHP code is **usually processed on a web server by a PHP interpreter** implemented as a module, a daemon or **a common gateway interface executable**.
- ☐ PHP has **in-built support for MySQL** which is one of the most widely used database management system.
- ☐ Various **web template systems, web content management systems** and **web frameworks** exist which can be employed to facilities the generation of that response.
- ☐ PHP files **can be embedded into HTML** with CSS, JavaScript, text, etc.



```
<!DOCTYPE>
<html>
<body>
<?php
echo "<h2>Hello First PHP</h2>";
?>
</body>
</html>
```

Programming Language – PHP Features (1)

Performance

- PHP script is executed **much faster than other scripts** such as JSP and ASP. **PHP uses its own memory**, so the server workload and loading time is automatically reduced, which results in faster processing speed and better performance.

Open Source

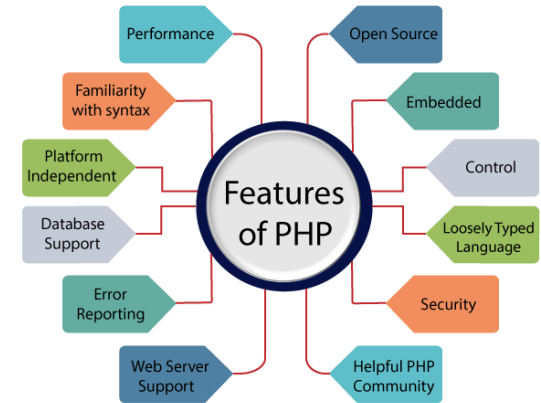
- PHP source code and software are **freely available on the web**. Any programmer can develop all the versions of PHP without paying any cost. All its components are **free to download and use**.

Familiarity with syntax

- PHP has **easily understandable syntax**. Programmers are comfortable coding with it.

Embedded

- PHP code can be easily **embedded within HTML tags and script**.



Source: javaTpoint

Programming Language – PHP Features (2)

Error reporting

- PHP has **predefined error reporting constraints** to generate an error notice or warning at runtime, e.g., E_ERROR, E_WARNING, E_STRICT, etc.

Loosely typed language

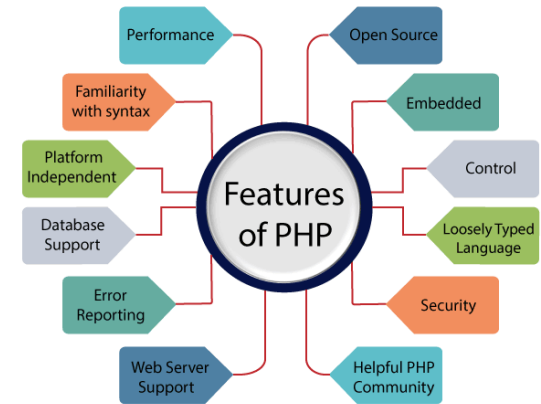
- PHP allows to **use a variable without declaring its datatype**. It will be automatically at the time of execution based on the type of data contains on its value.

Web server support

- PHP is **compatible with almost all local server** used today like Apache, Nginx, Microsoft IIS, etc.

PHP community

- It has a **large community of developers** who regularly updates **documentation, tutorials, online help** and **FAQs**. Learning PHP from the communities is one of the significant benefits.



Source: javaTpoint

Programming Language – PHP Frameworks

- ❑ **A PHP framework is a platform** which provides a **structure to develop web application**.
- ❑ These frameworks **save development time, stop rewriting the repeated code** and **provide rapid application development**.
 - Laravel framework
 - CodeIgniter framework
 - Zend framework
 - Slim framework
 - Phalcon framework
 - CakePHP framework
 - Symfony framework
 - FuelPHP framework

...



Source: javaTpoint



Question – Which programming languages are?



(A) `System.out.println(message);`

(H) `echo "Hello, World";`

1) Python

(B) `print("Hello World")`

(I) `DISPLAY "Hello, world!"`

2) C#

(C) `printf("hello, world\n");`

(J) `Write('Hello, world!')`

3) Java

4) Basic

(D) `std::cout << "Hello, world!\n";`

(K) `cat("Hello world\n")`

5) C

6) Cobol

(F) `Console.WriteLine("Hello, world!");`

(L) `puts("Hello World!");`

7) C++

8) Pascal

(G) `console.log("Hello World!");`

(M) `PRINT "Hello, World!"`

9) Python

10) Objective-C

11) JavaScript

12) R

Software and Software Engineering

Software Engineering

Needs of Software Engineering

☐ Huge Programming

- As the scale of software becomes extensive, it needs **systematic and disciplined approaches**.

☐ Adaptability

- If the software development is not based on engineering approaches, it is **difficult to adapt an existing one**.

☐ Cost

- The cost of software remains high if the proper process is not adapted.

☐ Dynamic Nature

- If the quality of the software is continually changing, **new upgrades need to be done** in the existing one with efficient approaches.

☐ Quality Management

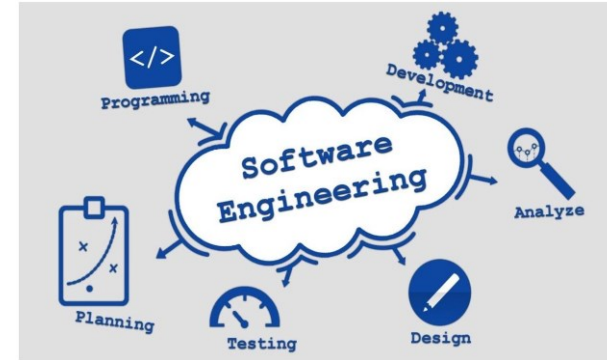
- Better procedure** of software development provides **a better and quality** software product.



Source: Brazilian Gringo

Definition of Software Engineering

- ❑ **Software engineering** is an **engineering discipline** that's applied to the software development in a **systematic approach**.
- ❑ The **IEEE** defines software engineering as
“The application of systematic, disciplined, quantifiable approach to the development, operations, and maintenance of software; that is, the application of engineering to Software”
- ❑ According to Wikipedia, software engineering is **the systematic application of engineering approaches** to the development of software.
- ❑ Software engineering is **an engineering branch associated with development of software product** using well-defined scientific principles, methods and procedures.

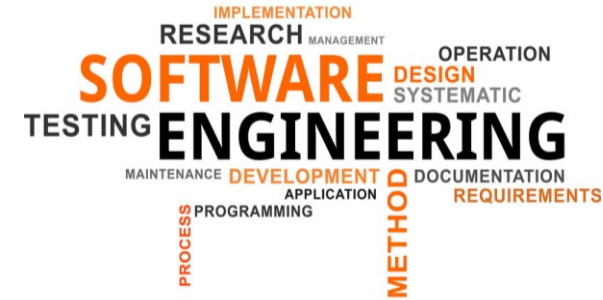


Source: Globashare



Software Engineering

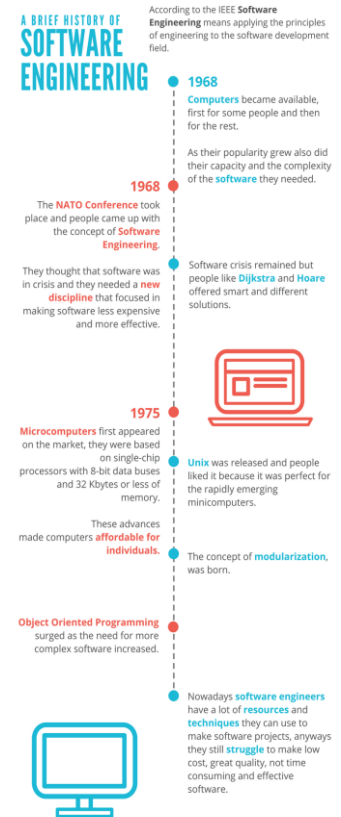
- ❑ **Software engineering** includes **theories, methods, and tools** to design and build a software.
- ❑ Software engineering helps to **meet the software specifications** cost-effectively, and to **ensure the quality**.
- ❑ Software engineering requires **various activities** to manage the project, developer tools, methods and theories that support the software production.
- ❑ Applying software engineering methods can result in **less expensive** and **more reliable software**.
- ❑ Software engineering is vital on **the long term to support any software changes** which requires the cost increase dramatically.
- ❑ Software engineering supports **different models, methods and techniques** for **different type of systems**.



Source: VectorStock

Software Engineering History

- ❑ In the 1940s, **computers** were invented and then **computing programming languages** were invented.
- ❑ In the 1950s, **large computer became available** to research institutions and universities for some scientific applications.
- ❑ In the late 1950s and earlier 1960s, **many programming languages** were introduced such as Fortran, Algol, Cobol, C, etc. However, **software became more complex and big, and hard to write.**
- ❑ In 1968 and 1969, the term, **software engineering**, was suggested at conference sponsored by NATO to discuss the '**software crisis**'.
- ❑ Throughout the 1970s and 1980s, a variety of new software engineering methods were developed, such as **structured programming, modularisation and object oriented programming**, etc.
- ❑ In the 1990s and 2000s, **the rise of the internet** led to very rapid growth in demand for software development. Software engineering is more important than ever before.



Source: [kenscourses.com](https://www.kenscourses.com)

☐ Software Requirements

- It is about the elicitation, analysis, specification, and validation of requirements for software. There are three different types: **functional requirements**, **non-functional requirement**, and **domain requirements**.

☐ Software Design

- It is about the process of defining the architecture, components, interfaces, and other characteristics of system. There are three different levels: **interface design**, **architecture design**, and **domain design**.

☐ Software Development

- It is about the activity of software construction including **programming**, **verification**, **debugging**.

☐ Software Testing

- It is about technical investigation conducted to provide stakeholders with information about the quality of the product or service.

☐ Software Maintenance

- It is about modifying and update software applications after shipping the software product.

Software Engineering Ethics

- ☐ Software engineering is carried out within **a social and legal frameworks** that may limit the freedom of works.
- ☐ Engineers must **behave in an ethical and morally responsible way** as a professional engineer.
- ☐ Professional societies and institutions such as **ACM** (Association of Computing Machinery), **IEEE** (Institute of Electrical and Electronic Engineers), and British Computer Society publish **a code of professional conduct** or **code of ethics**.
- ☐ Members of these organisation **sign up the code of practice**.
- ☐ Software engineering involves **wider responsibilities** than simply the application of technical skills.
- ☐ Some of ethics and professional responsibility are:
Confidentiality, Competence, Intellectual property rights, Computer misuse, etc.



Source: Swartz Campbell

Issues of Ethics and Professional Responsibility

☐ Confidentiality

- Engineers should normally **respect the confidentiality of their employers or clients** irrespective of whether or not a formal confidentiality agreement has been signed.

☐ Competence

- Engineers should be **not misrepresent their level of competence**. They should not knowingly accept work which is out of their competence.

☐ Intellectual property rights

- Engineers should **be aware of local laws governing the use of intellectual property** such as patents, copyright, etc. They should be careful to ensure that the intellectual property is protected.

☐ Computer misuse

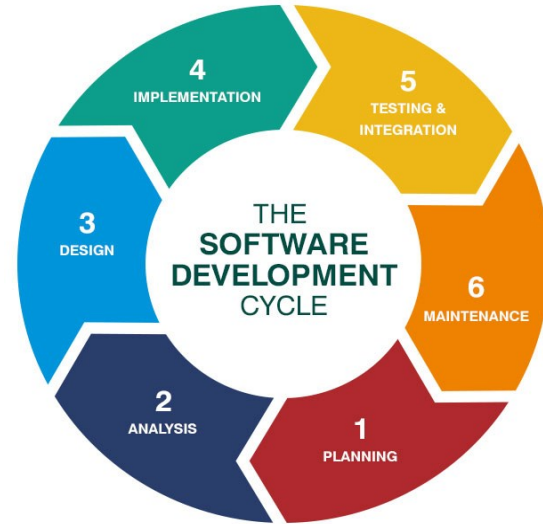
- Engineers should **not use their technical skills to misuse** other people's computers. Computer misuse ranges from relatively trivial (gaming on company resource) to extremely serious (virus dissemination).



Source: Peace Place Library

Software Development Life Cycle

- ❑ **Software Development Life Cycle (SDLC)** is the process of dividing software development work **into smaller, parallel or sequential steps** to improve software design, software management, software implementation, etc.
- ❑ SDLC is also known as **software development process, software development methodology**.
- ❑ SDLC aims to produce a **high-quality software** that meets or exceeds customer expectation **within times and cost estimates**.
- ❑ It typically consists of five to eight steps: **Planning, Analysis, Design, Implementation, Testing, Deployment, Maintenance, and Documentation**.
- ❑ There are various software engineering life cycle models defined and designed such as **Waterfall Model, V Model, Rapid Prototyping Model, Spiral Model, Agile Model**, etc.



Source: synotive.com

Benefits of Software Development Life Cycle

- ☐ With the SDLC, it is possible to clearly **see the goals and the problems** so that the plan is implemented with **precision** and **relevance**.
- ☐ A formal review is created at the end of each stage so that it allows to **have maximum management control**.
- ☐ The installation using the SDLC **has the necessary checks and balances** so that it will be tested with **precision before the installation**.
- ☐ The SDLC **provides a well-structured and well-documented paper** trail of the entire project with the records of everything that occurs.
- ☐ With a well-designed SDLC, **everything will be in order** so that a new project member can continue the process without complications.
- ☐ The SDLC makes **sticking to a project budget easier** with a well-organised plan so that it can see the clear **project timetables** and **costs**.
- ☐ The SDLC model **provides the project with flexibility** by having feed back into the earlier stages.

Software and Software Engineering

Software Engineer vs Developer

Software Developer vs Software Engineer

- ❑ A **software engineer** is a professional who **applies the principles of software engineering** for design, development, maintenance, test, and evaluation of computer software.
- ❑ A **software developer** is a professional who **build software** across various types of computer.



Source: technojobs

- ❑ The other key differences are:
 - Software engineer is **involved in the complete process** whereas software developer is **one aspect** of the software project building process
 - Software engineer **works with other components** of the hardware system whereas software developer write a **complete program**.
 - Software engineer is a **team activity** while software developer is **primarily a solitary activity**.
 - Software engineer **may create the tools** to develop software while software developers **use readymade tools** to build apps.
 - Software engineer trends to solve issues on a much **larger scale** whereas software developer trends to do everything that engineers do but on a **limited scale**.

Practices and Challenges for Software Engineer

Practices

- ☐ Software engineers should be **supportive of their colleagues**.
- ☐ Software engineers should meet the **highest professional standards** for software products.
- ☐ Software engineers should consider **ethical issues** for software development and maintenance.
- ☐ Software engineers should seek **helpful values** to the client as well as the employer.
- ☐ Software engineers should be able to **maintain integrity and independence** in their professional approach.

Challenges

- ☐ Because the cost of software failure is massive **in safety-critical areas** such as space, nuclear power plants, etc. **special extra attention** is required.
- ☐ **Increased market demand** is quite a challenge for software engineers
- ☐ **Dealing with the increased complexity of software** always demand **new applications and techniques**.
- ☐ The diversity of software systems typically **require communication and dependency** between systems.

Practices and Challenges for Software Developer

Practices

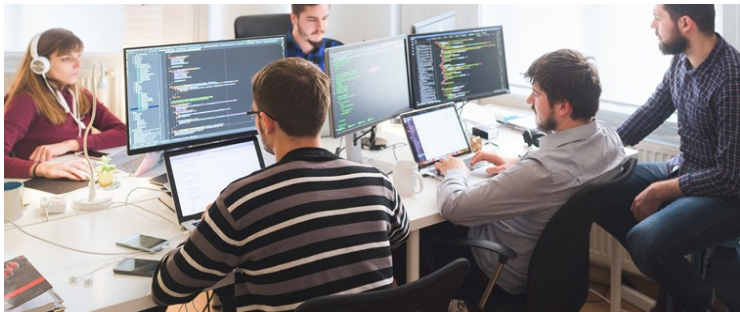
- ☐ Software developers need to **understand how the program code** helps the overall business.
- ☐ Software developers should **use the program code effectively** in the project.
- ☐ Software developers should **have your daily coding goals** for the project.
- ☐ Software developers **need to plan how to approach the coding task** in the project.
- ☐ Software developers should **not avoid the hardest program part** in the project.

Challenges

- ☐ **Misinterpreting end-user requirements** often happens.
- ☐ **Dealing with requirement changes** is always expected.
- ☐ Software is **difficult to implement** but **hard to maintain and extend**.
- ☐ **Late discovery of software important project flaws** has to be considered.
- ☐ **Poor quality of the software** is another common challenge which is often faced by a software especially in low-cost projects

Types of Software Developer and Engineer

- ❑ New types of software engineer and developer are required as **the software development landscape changes** constantly
- ❑ **The boundaries** between different types of software developers and engineers are **blurred**.
- ❑ The **same job title** might mean something completely different in **different organisations**.
- ❑ They are **often positioned together** on the job market.



Source: SEEK

- ❑ There are many different types of software developers and engineers:
 - Front End Developer
 - Back End Developer
 - Full Stack Developer
 - DevOps Engineer
 - Desktop Developer
 - Mobile Developer
 - Security Engineer
 - Data Scientist
 - Software Quality Assurance Engineer
 - Game Developer

...



Full Stack Developer

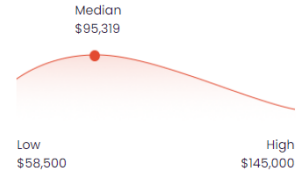
- ❑ **Full stack** refers to the collection of **a series of technologies** needed to complete a project but generally is known as **the combination of the frontend and backend**.
- ❑ **A full stack developer** is a software expert who is equally **proficient in frontend and backend development**.
- ❑ Full stack developers have **a broad skillset and extensive knowledge base** including HTML/CSS, JavaScript, Backend languages such as PHP, Python, Java, etc., Database, and Web architecture.
- ❑ Full stack developers also require to **understand many different libraries, frameworks and development environment**.
- ❑ Full stack developers need **years of experience in software development** to earn the title of full stack developer.

Full stack developer: Salary

\$95,319 / Year

Based on 24882 salaries

The average **full stack developer** salary in the USA is **\$95,319** per year or **\$48.88** per hour. Entry level positions start at **\$58,500** per year while most experienced workers make up to **\$145,000** per year.



Full stack developer: salaries per region

Massachusetts	\$115,000	<div></div>
New York	\$114,680	<div></div>
Virginia	\$114,256	<div></div>
New Jersey	\$107,250	<div></div>
Pennsylvania	\$107,250	<div></div>
Delaware	\$105,000	<div></div>

Show more

Source: [talent.com](https://www.talent.com)



DevOps Engineer

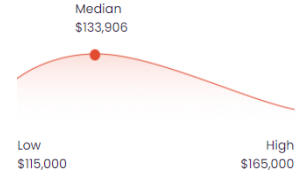
- ❑ **DevOps** is a set of practices that combines **software development (Dev)** and **IT operations (Ops)**.
- ❑ **A DevOps engineer is an IT professional** who works with software developers, system operators and admins, IT operations staff and others.
- ❑ A DevOps engineer introduces processes, tools, and methodologies to **balance needs throughout the software development life cycle**, from coding and deployment, to maintenance and updates.
- ❑ DevOps engineers need to know how to use and understand the roles of the following of tools: **Version control, Continuous Integration, Configuration management, Deployment automation, Containers, Monitoring and Analytics, Testing and Cloud Quality tools, Network protocols, Infrastructure orchestration**, etc.

Devops engineer: Salary

\$133,906 / Year

Based on 19959 salaries

The average **devops engineer** salary in the USA is **\$133,906** per year or **\$68.67** per hour. Entry level positions start at **\$115,000** per year while most experienced workers make up to **\$165,000** per year.



Devops engineer: salaries per region

California	\$146,762	<div></div>
Rhode Island	\$146,250	<div></div>
New York	\$145,000	<div></div>
Massachusetts	\$140,000	<div></div>
Hawaii	\$135,000	<div></div>
South Dakota	\$135,000	<div></div>

Show more

Source: [talent.com](https://www.talent.com)



Desktop Developer

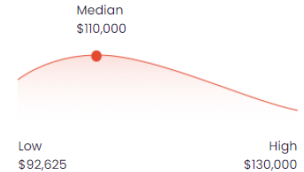
- ❑ **A desktop application** is a computer program that **runs locally on a computer device**, such as desktop or laptop computer, in contrast to a web application over the internet from a remote server.
- ❑ **A desktop developer** is a programmer who writes code for **software applications that run natively on operation systems** like macOS, Windows, and Linux.
- ❑ While web and mobile applications are become more and more capable and advanced, there are a number of reasons why desktop applications is still need: **Better performance, Working offline, Security limitations, Controlled environment, Hardware integration**, etc.
- ❑ Common desktop software development languages are: **C#** development, **.NET** development, **WPF/WinForms** development, **Swift** development, **C/C++** development, **Java** development, etc.

Desktop developer: Salary

\$110,000 / Year

Based on 138 salaries

The average **desktop developer** salary in the **USA** is **\$110,000** per year or **\$56.41** per hour. Entry level positions start at **\$92,625** per year while most experienced workers make up to **\$130,000** per year.



Desktop developer: salaries per region

California	\$120,000	<div></div>
North Carolina	\$99,450	<div></div>
Texas	\$98,750	<div></div>
Pennsylvania	\$93,750	<div></div>

Source: [talent.com](https://www.talent.com)

Mobile (App) Developer

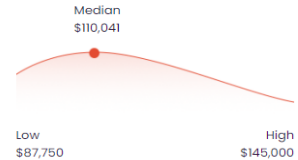
- ☐ A **mobile application**, also referred as **mobile app**, is **designed to run on mobile device** such as phone, tablet, or watch.
- ☐ A mobile developer **specialises in mobile technology** such as building apps for **Google's Android**, **App's iOS** and Microsoft's Windows Phone platforms.
- ☐ Mobile developers have to learn the programming languages and development environment **for their chosen platform**.
- ☐ There are few major platforms, each with its own core language and development environment such as **Java for Android**, **Swift for iOS** and **C# for Windows phone**.
- ☐ **Hybrid mobile development** is getting popular to create a single app that can **run on multiple platforms** including Ionic, Xamarin, PhoneGap, React Native, Flutter,

Mobile Developer: Salary

\$110,041 / Year

Based on 10648 salaries

The average **Mobile Developer** salary in the USA is **\$110,041** per year or **\$56.43** per hour. Entry level positions start at **\$87,750** per year while most experienced workers make up to **\$145,000** per year.



Mobile Developer: salaries per region

Massachusetts	\$126,750	<div></div>
New York	\$125,000	<div></div>
California	\$122,416	<div></div>
North Carolina	\$121,875	<div></div>
Pennsylvania	\$117,000	<div></div>
Washington	\$115,000	<div></div>
Delaware	\$18,525	<div></div>

Show more

Source: [talent.com](https://www.talent.com)



Security Engineer

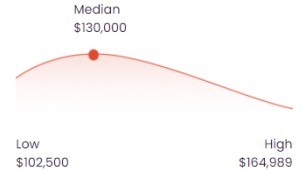
- ❑ A security engineer is responsible for **testing and screening security software**, and **monitoring networks and systems** to check security breaches or intrusions.
- ❑ Security engineers usually **works as part of a larger technology team** and report directly to upper management.
- ❑ **Some of the duties and responsibilities are:**
 - Developing a set of security standards and practices.
 - Recommending security enhancements to management.
 - Installing firewalls and data encryption programs.
 - Monitoring networks and systems for security.
 - Watching out for irregular system behaviour.
 - Supervising changes in software, hardware and user needs.
 - Leading incident response activities.

Security engineer: Salary

\$130,000 / Year

Based on 18424 salaries

The average **security engineer** salary in the USA is **\$130,000** per year or **\$66.67** per hour. Entry level positions start at **\$102,500** per year while most experienced workers make up to **\$164,989** per year.



Security engineer: salaries per region

New York	\$146,250	<div></div>
Massachusetts	\$140,000	<div></div>
California	\$140,000	<div></div>
Maine	\$136,500	<div></div>
Wyoming	\$136,500	<div></div>
North Dakota	\$136,500	<div></div>

Show more

Source: [talent.com](https://www.talent.com)



Data Scientist

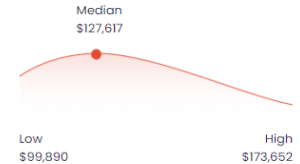
- ❑ **Data Science** is the field of study that combines domain expertise, programming skills, and knowledge of mathematics and statistics **to extract meaningful insights from data.**
- ❑ A data scientist is a professional responsible for **collecting, analysing, and interpreting** extremely large amounts of data.
- ❑ The types of data could be **structured, unstructured** and **semi-structured data** collected from various resources.
- ❑ The job for a data scientist **requires the use of advanced analytics technologies**, including machine learning and deep learning, predictive modelling, etc.
- ❑ **The demand** for data science skills has **grown significantly over the years** as companies look to glean useful information from big data.

Data Scientist: Salary

\$127,617 / Year

Based on 10000 salaries

The average **Data Scientist** salary in the USA is **\$127,617** per year or **\$65.44** per hour. Entry level positions start at **\$99,890** per year while most experienced workers make up to **\$173,652** per year.



Data Scientist: salaries per region

Nebraska	\$195,000	<div><div></div></div>
New York	\$150,000	<div><div></div></div>
California	\$150,000	<div><div></div></div>
Nevada	\$145,000	<div><div></div></div>
Oregon	\$137,500	<div><div></div></div>
Iowa	\$136,000	<div><div></div></div>
Montana	\$47,629	<div><div></div></div>

Show more

Source: [talent.com](https://www.talent.com)

Software Quality Assurance Engineer

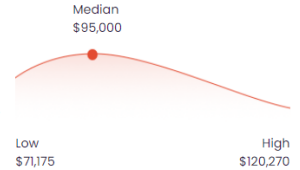
- ❑ Software delays are costly for a company, so it is important to **meet target date and stay within budget**.
- ❑ **A software quality assurance (QA) engineer** monitors every phase of the development process to ensure that the design and software adhere to **company standards**.
- ❑ **Some of the duties and responsibilities are:**
 - Discovering bugs within software.
 - Performing and documenting risk analysis.
 - Creating test plans.
 - Developing a set of security standards and practices.
 - Identifying any potential problems that users might encounter.
 - Reviewing user interface for consistency and functionality.
 - Driving innovation and streamline overall testing processes.

Software qa engineer: Salary

\$95,000 / Year

Based on 795 salaries

The average **software qa engineer** salary in the USA is **\$95,000** per year or **\$48.72** per hour. Entry level positions start at **\$71,175** per year while most experienced workers make up to **\$120,270** per year.



Software qa engineer: salaries per region

Massachusetts	\$110,000	Based on 67 salaries
California	\$104,899	
New Jersey	\$100,000	
Michigan	\$100,000	
Pennsylvania	\$100,000	
Colorado	\$97,500	

Show more

Source: [talent.com](https://www.talent.com)

- ☐ **Program** is a set of instructions telling a computer **how to work** and **Application** is a program or group of programs performing a specific task **for end-users**.
- ☐ Depending on the purpose of software, there are **a number of software types** including stand alone software, embedded software, data collection software, systems of system, etc.
- ☐ **Software failure** occurs when software does **not provide the expected result** with respect to specification input values.
- ☐ Good software provides **maintainability, dependability, efficiency and accessibility**.
- ☐ **Thousands of different programming** languages are available for software development, including Python, Java, JavaScript, C#, C++, PHP, etc.
- ☐ A developer can **easily maintain the workflow** of the project using the development tools.

SUMMARY

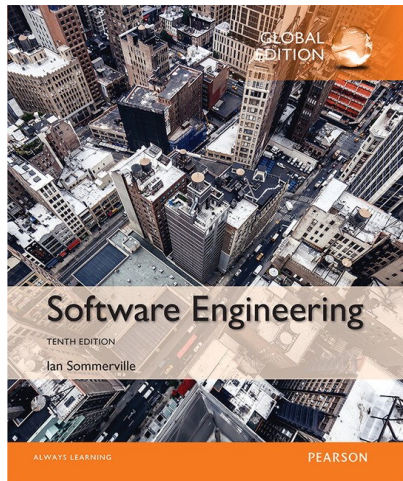
Source: VectorStock

Summary

- ☐ There are **many reasons why software engineering is required** for software development such as cost, time, quality, etc.
- ☐ Software engineering helps to meet **the software specifications** cost-effectively, and to **ensure the quality**.
- ☐ Engineers must **behave in an ethical and morally responsible way** as a professional engineer.
- ☐ **Software Development Life Cycle (SDLC)** is the process of dividing software development work into smaller, parallel or sequential steps to improve design, product management, and project management.
- ☐ There are **many different types of software developers and engineers**, including full stack developer, DevOps engineer, desktop developer, mobile developer, security engineer, etc.

SUMMARY

Source: VectorStock



Software Engineering


Ian Sommerville

Introduction

codegiant ARTICLES | DISCOVER CODEGiant

Software Development Life Cycle — The Ultimate Guide [2020]

Team Codegiant Follow
Apr 21, 2020 · 40 min read



10min30sec44

The software development life cycle is about producing high-quality software promptly.

And your team needs to follow the software development life cycle steps to achieve that.

It's also essential that your team adheres to the proper SDLC model.

This guide will be like the ultimate tutorial to SDLC.

We'll start by "What's the definition of SDLC" and then transition to the different phases of the software development life cycle and its models.

codegiant


Team Codegiant

Software Development Life Cycle – The Ultimate Guide

javaTpoint

Home Python Java JavaScript HTML SQL PHP C# C++ TS Aptitude Reasoning Selenium DBMS

For Videos Join Our Youtube Channel: Join Now




Feedback

Send your feedback to feedback@javatpoint.com

100+ Latest Updates

- 17th Aug - MongoDB \$group operator
- 17th Aug - MongoDB Regexp
- 17th Aug - MongoDB \$eq Operator
- 17th Aug - MongoDB \$unwind operator
- 17th Aug - Mate (Polygon) Cryptocurrency
- 17th Aug - Pi Cryptocurrency Value in INR
- 17th Aug - Download Music from YouTube
- 17th Aug - Parsephone for Me
- 17th Aug - Neuro-linguistic programming
- 17th Aug - How to download YouTube videos on Computer

Latest Tutorials



javaTpoint

Python, Java, JavaScript, C#, C++, PHP

OTHER REFERENCE

- Wikipedia:
 - https://en.wikipedia.org/wiki/Software_engineering
 - https://en.wikipedia.org/wiki/Mobile_app_development
 - https://en.wikipedia.org/wiki/Embedded_software
 - https://en.wikipedia.org/wiki/Data_collection_system
 - https://en.wikipedia.org/wiki/Programming_language_generations
 - <https://en.wikipedia.org/wiki/DevOps>
- Guru99: <https://www.guru99.com/difference-software-engineer-developer.html>
- GeeksforGeeks: <https://www.geeksforgeeks.org/difference-between-software-and-application/>
- Codecademy: <https://www.codecademy.com/resources/blog/programming-languages/>
- Software Testing Help: <https://www.softwaretestinghelp.com/software-development-tools/>
- JavaTpoint: <https://www.javatpoint.com/software-engineering-tutorial>
- OmarElgabry's Blog: <https://medium.com/omarelgabrys-blog/software-engineering-introduction-part-1-b79238ec97ee>
- The balancecareers: <https://www.thebalancecareers.com/list-of-software-quality-assurance-qa-engineer-skills-2062484>
- SearchEnterpriseAI: <https://searchenterpriseai.techtarget.com/definition/data-scientist>
- CareerExplorer: <https://www.careerexplorer.com/careers/security-engineer/>

THANKS!

Any questions?
y.s.park@warwick.ac.uk