**ASSIGNMENT 1 – MODULE 1 & 2**

1. **Write, in your own words, a brief (one paragraph) summary of its distinguishing features, translation methods, as well as its historical relationship with other languages that preceded or followed it**

**C, C++, Java, Python.**

**C language:**

**Distinguished features:**

C programs are capable of being written on one platform and being run on another with or without any modification. Modularity or structured language concept in C allows the program to be broken into smaller units and run individually with the help of functions. Since it is a compiler-based language, it is much faster than other programming languages like Java or Python, which are interpreter based. C language comprises of its library which has a wide range of built-in functions. It supports the feature of DMA (Dynamic Memory Allocation), which helps in the utilization and management of memory and it is also an unique feature from other languages.

**Translational methods:**

The source file input is mapped to the source character set. Continuation lines are spliced with the next line. The source code is parsed into whitespace and preprocessing tokens. The preprocessor is applied, which executes directives, expands macros, and applies pragmas. All preprocessor related directives are then deleted. Source character set values in character constants and string literals are mapped to the execution character set. String literals adjacent to each other are concatenated. The source code is parsed into tokens, which comprise the translation unit. External references are resolved, and the program image is formed.

**Historical relationships with other languages:**

The origin of C is closely tied to the development of the Unix operating system, originally implemented in assembly language on a PDP-7 by Dennis Ritchie and Ken Thompson. At first, they tried to make a Fortran compiler, but soon gave up the idea. Instead, he created a cut-down version of the recently developed BCPL systems programming language. The official description of BCPL was not available at the time and Thompson modified the syntax to be less wordy, producing the similar but somewhat simpler B. However, few utilities were ultimately written in B because it was too slow, and B could not take advantage of PDP-11 features such as byte addressability. In 1972, Ritchie started to improve B, which resulted in creating a new language C. Many later languages have borrowed directly or indirectly from C, including C++, C#, Unix's C shell, D, Go, Java, JavaScript, Julia, Limbo, LPC, Objective-C, Perl, PHP, Python, Ruby, Rust, Swift, Verilog and SystemVerilog. These languages have drawn many of their control structures and other basic features from C. Most of them express highly similar syntax to C, and they tend to combine the recognizable expression and statement syntax of C with underlying type systems, data models, and semantics that can be radically different.

**C++ language:**

**Distinguished features:**

The simple context of C++ gives an appeal to programmers, who are eager to learn a new programming language. C++ is a high-level object oriented programming language making it easier for the user to work as we can closely associate it with the human-comprehensible language. It is a case sensitive language which treats the uppercase and lowercase characters in a different manner. It is a strongly tight syntax based programming language.

**Translational methods:**

The translation unit of C++ is much similar to that of C which is discussed above.

**Historical relationships with other languages:**

C++ was initially known as “C with classes, ” and was renamed C++ in 1983. The trace of the programming language C++ can be done back to 1979 when Bjarne Stroustrup was doing some development for this thesis for PhD and he had the chance to figure with, a language called Simula primarily designed for simulations. The first C++ with categories compiler was referred to as Cfront, that got derived from a C compiler referred to as CPre. C++ is still the third most popular programming language according to the TIOBE Index’s latest study of 2020, behind Java & C, by knocking out Python. C++ finds a wide range of applications – from GUI applications to 3D graphics for games to real-time mathematical simulations next to Java.

**Java language:**

**Distinguished features:**

With Java's secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption. Java compiler generates an architecture-neutral object file format, which makes the compiled code executable on many processors, with the presence of Java runtime system. Java makes an effort to eliminate error-prone situations by emphasizing mainly on compile time error checking and runtime checking. With the use of Just-In-Time compilers, Java enables high performance. Java is designed for the distributed environment of the internet. With Java's multithreaded feature it is possible to write programs that can perform many tasks simultaneously. This design feature allows the developers to construct interactive applications that can run smoothly. Java is considered to be more dynamic than C or C++ since it is designed to adapt to an evolving environment.

**Translational methods:**

Here the translation is targeted to the Java Byte Code language rather than assembly or machine language. The Byte Code undergoes its own translation process (including many of the same stages) when the program is run. During the translation phase each type of data in a JSP page is treated differently. Static data is transformed into code that will emit the data into the response stream. Directives are used to control how the web container translates and executes the JSP page. Scripting elements are inserted into the JSP page’s servlet class. Expression language expressions are passed as parameters to calls to the JSP expression evaluator. Property elements are converted into method calls to JavaBeans components. Server elements are converted into invocations of the Java Servlet API. Plugin elements are converted into browser-specific markup for activating an applet. Custom tags are converted into calls to the tag handler that implements the custom tag.

**Historical relationships with other languages:**

Java was developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle) and released in 1995 as a core component of Sun Microsystems' Java platform. Java was originally designed for interactive television, but it was too advanced for the digital cable television industry at the time. Currently, Java is used in internet programming, mobile devices, games, e-business solutions, etc. Java is a statically typed object-oriented language that uses a syntax similar to (but incompatible with) C++. The minimum of code needed to compile for C++ is a function, for Java is a class. C++ can allocate arbitrary blocks of memory but Java only allocates memory via object instantiation. Gosu is an extensible type-system language compiled to Java bytecode. Join Java is a language that extends Java with join-calculus semantics. Processing is a visualization and animation language and framework based on Java-like syntax. Beanshell is a scripting language whose syntax is close to Java. Ateji PX is an extension of Java for easy parallel programming on multicore, GPU, Grid and Cloud.

**Python language:**

**Distinguished features:**

Python is very easy to code as compared to other popular languages like Java and C++. Anyone can learn Basic Python syntax in just a few hours. Thus, it is programmer-friendly. Python provides us with a variety of constructs that help us focus on the solution rather than on the syntax. Graphical User interfaces can be made using a module such as PyQt5, PyQt4, wxPython, or Tk in python. Python is an extensible and easily integrated language. We can write us some Python code into C or C++ language and also we can compile that code in C/C++ language. Python has a large standard library which provides a rich set of module and functions so you do not have to write your own code for every single thing. Python is freely available and you can download it from the Python Official Website. Also, it is open-source. This means that its source code is available to the public so that they can download it, change it, use it, and distribute it. Python is dynamically-typed and the type for a value is decided at runtime, not in advance. So we don’t need to specify the type of data while declaring it.

**Translational methods:**

To translate the characters in the string translate() is used to make the translations. This function uses the translation mapping specified using the maketrans(). Translation can also be achieved by specifying the translation dictionary and passing as an object which acts as a mapping. In this case, there is no need for maketrans() to perform translations.

**Historical relationships with other languages:**

ABC language and Modula 3 are the two languages contributed to Python as a programming language. Guido van Rossum began working on Python in the late 1980s, as a successor to the ABC and first released it in 1991 as Python 0.9.0. Python 2.0 was released on 16 October 2000 with many major new features, including a cycle-detecting garbage collector and support for Unicode. Python's design offers some support for functional programming in the Lisp tradition. One of the cross compilers Nuitka compiles Python into C++. Google's Grumpy (latest release in 2017) transpiles Python 2 to Go. Iron Python allows running Python 2.7 programs on the .NET Common Language Runtime. Jython compiles Python 2.7 to Java bytecode, allowing the use of the Java libraries from a Python program. MyHDL is a python-based hardware description language that converts MyHDL code to Verilog or VHDL code. Brython, Transcrypt and Pyjs (latest release in 2012) compile Python to JavaScript.

Finally Python has a clear syntax and doesn’t require as many lines of code as Java or C to give you comparable results.

1. **Find the new version on the Web. What new features does this version add to old version? What old features of the programming language are eliminated by the newer version, if any? C, C++, Java, Python**

**For C:**

C17 is the most recent standard for the C programming language, prepared in 2017 and published in June 2018. It replaced C11 and it was superseded by C2x which was expected to be released after December 2021.

Some of its proposed features:

* Single-argument \_Static\_assert.
* C++11 style attribute syntax.
* Extended binary floating-point arithmetic, decimal floating-point arithmetic
* memccpy(), strdup(), strndup() – similar to functions found in the POSIX and SVID C extensions
* char8\_t type, in the same vein as char16\_t and char32\_t added in C11.
* To add a new principle to the "original principles" of C: "15. Application Programming Interfaces (APIs) should be self-documenting when possible. In particular, the order of parameters in function declarations should be arranged such that the size of an array appears before the array. The purpose is to allow Variable-Length Array (VLA) notation to be used. This not only makes the code's purpose clearer to human readers, but also makes static analysis easier. Any new APIs added to the Standard should take this into consideration."
* Possibly IEEE 754-2019, with fmaxmag / fminmag removed.
* Standardization of the typeof(...) operator.

Some of the old features are:

* Multi-threading support for atomic operations supporting the C11 memory model.
* Improved Unicode support based on the C Unicode Technical Report ISO/IEC TR 19769:2004.
* Removal of the gets function which was deprecated in the previous C language standard revision.
* More macros for querying the characteristics of floating-point types, concerning subnormal floating-point numbers and the number of decimal digits the type is able to store.
* Anonymous structures and unions, useful when unions and structures are nested.
* Static assertions, which are evaluated during translation at a later phase than #if and #error, when types are understood by the translator.
* The quick\_exit function as a third way to terminate a program, intended to do at least minimal deinitialization if termination with exit fails.
* A new timespec\_get function and corresponding structure in <time.h> with a degree of POSIX compatibility.

**For C++:**

C++ 17 is the most recent version of C++ programming language revised by ISO/IEC 14882 standard.

Existing C++ features removed:

1. Removal of Trigraphs.
2. Removal of Registers.
3. Removal of Deprecated operator ++.
4. Removal of Deprecated Exception Specifications.
5. Removal of auto\_ptr.

New C++ 17 features:

1. Perfomance fixes.
2. Language clarification.
3. Templates and attributes.
4. Simplification.
5. Library changes – Filesystem.
6. Library changes – Parallel algorithms.
7. Library changes – Utils
8. Searchers.

Proposed features of C++ 20:

* Three-way comparison using the "spaceship operator", operator <=>
* Initialization of an additional variable within a range-based for statement.
* Lambdas in unevaluated contexts.
* Default constructible and assignable stateless lambdas.
* String literals as template parameters.
* Atomic smart pointers (such as std::atomic<shared\_ptr<T>> and std::atomic<weak\_ptr<T>>).
* std::to\_address to convert a pointer to a raw pointer
* Removing the need for typename in certain circumstances
* New standard attributes [[no\_unique\_address]], [[likely]] and [[unlikely]]
* Calendar and time-zone additions to <chrono>
* std::span, providing a view to a contiguous array.

**For Java:**

As of March 2021, the latest version is Java 16, with Java 11, a currently supported long-term support (LTS) version, released on September 25, 2018. Oracle released the last zero-cost public update for the legacy version Java 8 LTS in January 2019 for commercial use, although it will otherwise still support Java 8 with public updates for personal use indefinitely. Oracle (and others) highly recommend uninstalling outdated versions of Java because of serious risks due to unresolved security issues. Since Java 9, 10, 12, 13, 14, and 15 are no longer supported, Oracle advises its users to immediately transition to the latest version (currently Java 16) or an LTS release.

Features included:

* Vector API.
* Elastic Metaspace.
* Enable C++14 language features.
* Windows/AArch64 Port.
* Concurrent Thread-Stack Processing.
* Foreign Linker API (Incubator).
* Warnings for Value-Based Classes.
* Strongly encapsulate JDK Internals by default.

Features removed:

* strictfp keyword.
* Concurrency utilities in package.
* JavaDB was removed from JDK.
* Removal of Native header generation tool (javah).
* Removal of Nashorn Javascript engine.
* Removal of Pack200 tools and API.
* Removal of Concurrent Mark Sweep (CMS) Garbage collector.
* Removal of Solaris and SPARC ports.

Proposed features to be included in Java SE 17 (future release on September 2021)

* Enhanced Pseudo-Random Number Generators.
* New macOS Rendering Pipeline.
* macOS/AArch64 Port.
* Deprecate the Applet API for Removal.
* Improved interoperability with native code, to enable Java source code to call functions and use data types from other languages, in a way that is easier and has better performance than today.
* Virtual threads are mapped to OS threads in a many-to-many relationship, in contrast to the many-to-one relationship from the original green threads implementation in early versions of Java.

**For python:**

Python 3.9.2 is the latest version of Python came into existence on February 2021.

New features:

* Union operators added to dict.
* Relaxed grammar restrictions on decorators.
* String methods to remove prefixes and suffixes.
* Flexible function and variable annotations.
* CPython now uses a new parser based on PEG.
* Fast access to module state from methods of C extension types.
* The IANA Time Zone Database is now present in the standard library in the zoneinfo module.
* An implementation of a topological sort of a graph is now provided in the new graphlib module.

Existing features modified:

* The dict type has been reimplemented to use a more compact representation.
* Customization of class creation has been simplified with the new protocol.
* The default settings and feature set of the ssl module have been improved.
* Removal of the Python 2 input function, and the renaming of the raw\_input function to input.
* Moving reduce out of the built-in namespace and into func tools.
* Adding support for optional function annotations that can be used for informal type declarations.
* Adding support for optional function annotations that can be used for informal type declarations.
* A change in integer division functionality.

Proposed features in Python 3.10 (future release on April 2021):

* Parenthesized context managers.
* Better error messages in the parser.
* Precise line numbers for debugging and other tools.
* Structural Pattern Matching.
* Patterns with a literal and variable.
* Patterns with positional parameters.
* New Type Union Operator.
* TypeAlias Annotation.

1. **Consider the following quote by L.Flon:**

**“There does not now, nor will there ever exist, a programming language in which it is the least bit hard to write bad programs.”**

**Are programming languages inherently flawed? Or is he suggesting that programmers are inherently inept? Or is there a middle- ground interpretation? Explain.**

There is one flaw that all of the top 20 languages in the popularity rankings have: They can only run in one direction, forward, like a car without a reverse gear. All the top 20 languages operate the same way. We have our source code written in that language, which gets massaged and processed down to the machine language that runs on the hardware. Each part of the program when executed causes other parts of the code to be executed, all the while changing the state of the data associated with the code.

When writing the code, the programmer is supposed to anticipate every single branch and possible reaction to the user’s inputs or the data being processed. During operation of the code, the program follows an unexpected path, or bad data causes it to malfunction, and this results in a bug that has to be fixed. In order to fix the bug, the programmer has to back up in his/her mind the path that was followed to find out where the deviation from the planned route occurred.

Today, most programmers have discovered that machines are much easier to understand than humans. Most machines work on a strict, known, and consistent logic. The machine's characteristics and responses are accurately (more or less) described in the installation, operation, and maintenance documentation. If the machine doesn't do what you asked (programmed) it to do, you can follow the debugging procedures, find the problem, fix the problem, and be done with it.

1. **Discuss about the design issues with respect to readability, writability, reliability and cost for any three imperative programming languages**







