1. Evolution of programming languages:

1840 – Analytical Engine Code
The Analytical Engine was a theoretical (i.e., never built) mechanical general-purpose computer, created by British mathematician Charles Babbage. Ada Lovelace came across the idea, and created some code for the Analytical Engine. That’s why she’s considered the first programmer ever.

1943 – ENIAC Coding System
The ENIAC is regarded as the first electronic general-purpose computer. Both the computer and its coding were created by John von Neumann, John Mauchly, and J. Presper Eckert.

1949 – Brief Code (Later Short Code)
Initially proposed by John Mauchly, it was one of the first attempts of an assembly language.

**Autocode (1952):**This family of “simplified coding systems” was created in the 1950s specifically for use with the digital computers at the universities of Manchester, Cambridge and London. Considered by many to be the first complied programming language ever invented, Autocode was developed by Alick Glennie to be both comprehensible and high-level.

1954 – Fortran
One of the most popular high-level programming languages. It was created by John W. Backus at IBM as an easier alternative to programming in assembly.

1958 – LISP
Created by John McCarthy, one of the pioneers of AI as well.

1959 – COBOL
The name stands for COmmon Business-Oriented Language, as the language was aimed mainly at banks, financial institutions and companies.

1964 – BASIC
Beginner’s All-purpose Symbolic Instruction Code, a family of general-purpose, high-level programming languages whose design philosophy emphasizes ease of use.

**Algol 68 (1968):**Short for Algorithmic Language 1968, Algol 68 was an imperative programming language designed as a successor to Algol 60. With a wider scope of application and rigorously defined syntax, this language was the first to be fully defined before it was implemented.

1970 – Pascal
Pascal is an influential imperative and procedural programming language, designed in 1968–1969 and published in 1970 by Niklaus Wirth as a small and efficient language intended to encourage good programming practices using structured programming and data structuring.

1972 – Smalltalk
The language that started to inflate the popularity of object-oriented programming.

1972 – C
Created by Dennis Ritchie and Ken Thompson at the AT&T Bell Labs. It’s simplicity and efficiency made it one of the most popular languages around the world.

1972 – SQL
Created at IBM, it became the standard for dealing with databases.

1983 – C++
Originally named “C With Classes”, it brought object-orientation to C (which is technically a subset of C++).

1987 – Perl
Perl is a family of high-level, general-purpose, interpreted, dynamic programming languages.

1991 – Python
A high-level language that emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than would be possible in languages such as C.

1995 – Java
Java is the most popular object-oriented programming language around, and it was created to have as few implementation dependencies as possible. It’s widely used in commercial and business applications.

**PHP (1995):**PHP is a server-side scripting language used for both Web development and general-purpose programming. This language can be mixed with HTML code or used in combination with templating engines and Web frameworks. It was originally not meant to be a programming language, but grew organically over time.

**C# (2001):**This multi-paradigm programming language was developed by Microsoft within the .NET framework. It was intended to be simple, modern and object-oriented. The most recent version of C# was released in 2012.

**Visual Basic .NET (2001):**A successor to the original Visual Basic language, Visual Basic .NET is a high-level programming language implemented on the .NET framework. It uses statements to specify actions and is one of the two main languages targeting the .NET framework, along with Visual C#.

**Swift (2014):**Swift was created by Apple for iOS and OS C development. It was introduced in 2014 at Apple’s Worldwide Developers Conference. Designed to work with Apple’s Cocoa and Cocoa Touch frameworks, Swift is meant to be more concise and resilient to erroneous code.

1. Distinguish between modular programming paradigm and object oriented programming paradigm:
2. Modular Programming Paradigm:It is also known as 'stepwise refinement' or 'top-down design' paradigm. It is similar to the procedural programming. In modular programming, procedures of a common functionality are grouped together into separate modules and the main program coordinates calls to procedures in separate modules and hands over data as parameter. It can also be described as a software design technique that emphasizes separating the functionality of a program into independent, interchangeable modules, such that each contains everything necessary to execute only one aspect of the desired functionality.
3. Object Oriented Programming Paradigm(OOP): in OOP, a program is structured as a well of interacting objects and these objects interact by sending messages to each other. It is an approach to the overall organization of a program. Data is hidden in OOP. OOP contains different types of objects, each corresponding to a complex real world objects or any complex data or a concept such as a bank customer, a bank account or any departmental store.



fig1: diagram of modular programming paradigm



fig2: diagram of object oriented programming paradigm