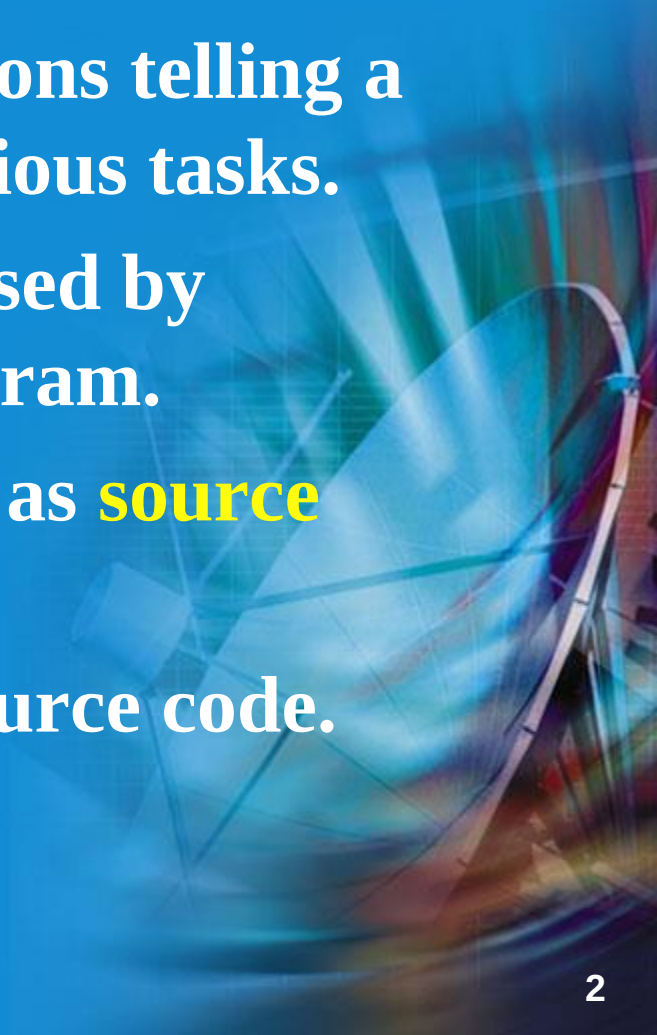
Programming Concepts and Languages

Programming Concepts

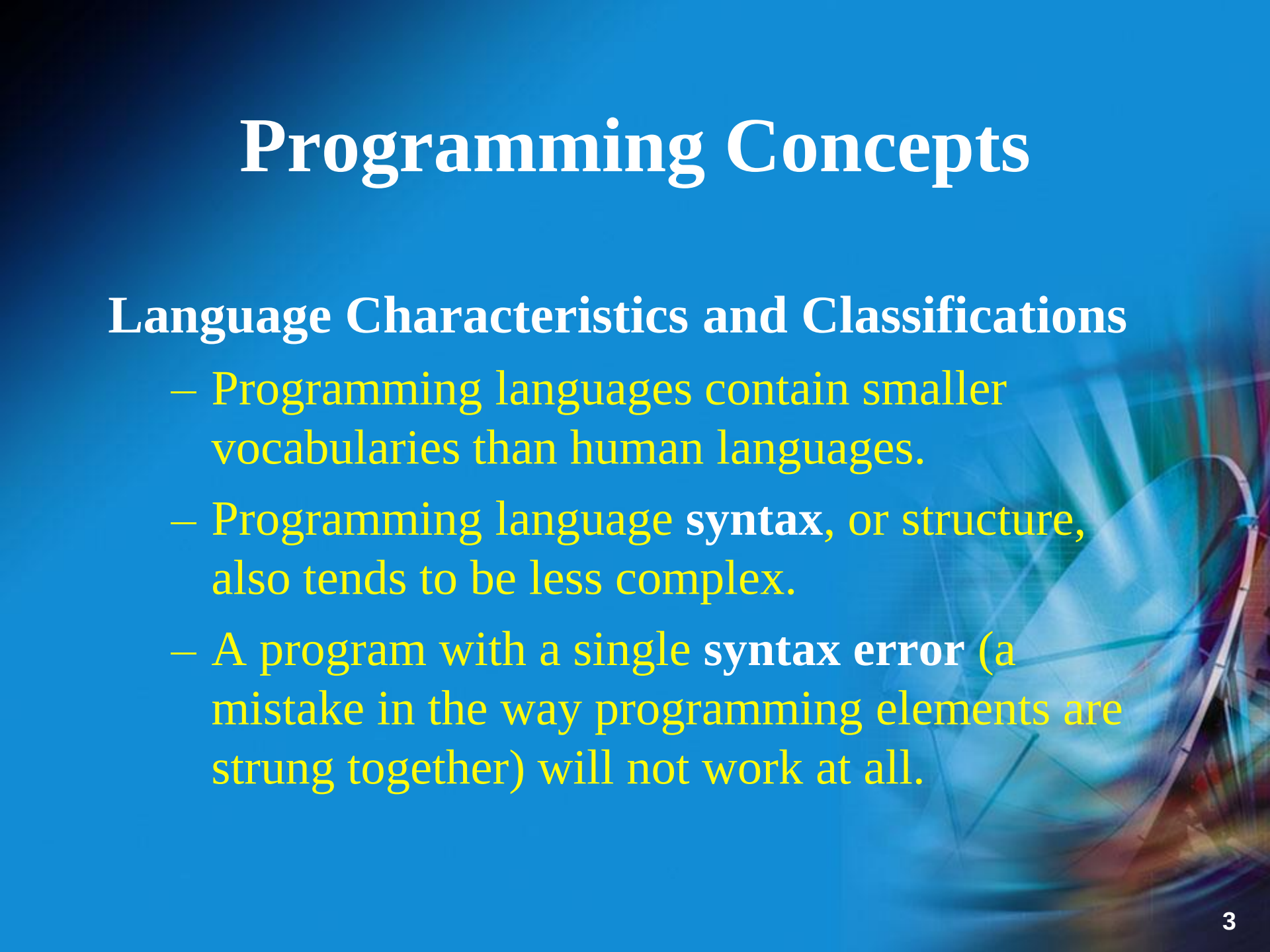
- A **program** is a set of instructions telling a computer how to perform various tasks.
- A **programming language** is used by programmers to create a program.
- Programs are also referred to as **source code**.
- **Coding** is the act of writing source code.



Programming Concepts

Language Characteristics and Classifications

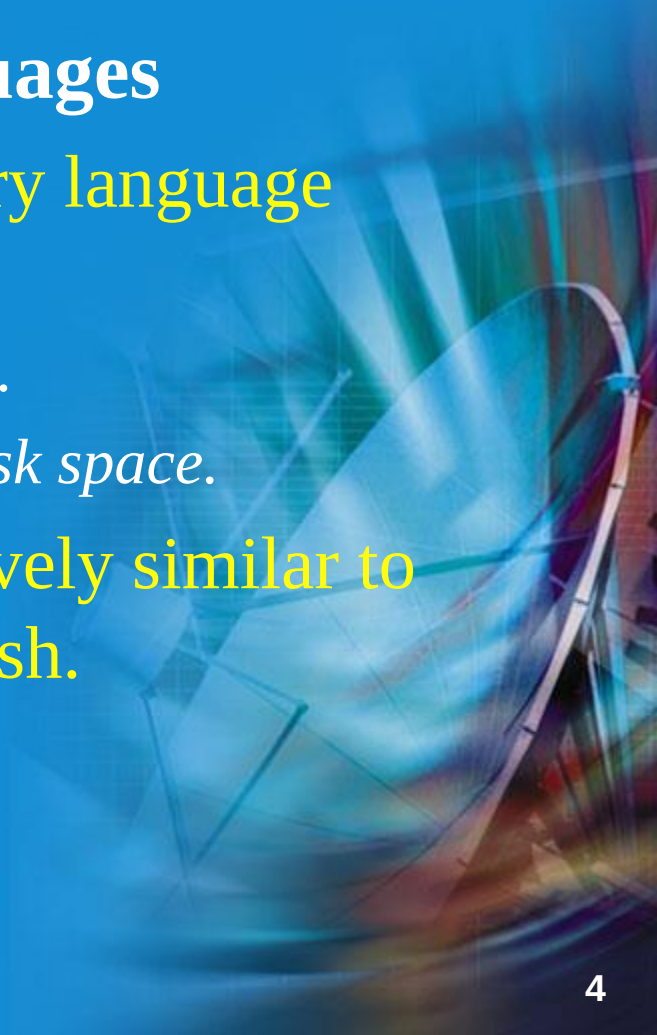
- Programming languages contain smaller vocabularies than human languages.
- Programming language **syntax**, or structure, also tends to be less complex.
- A program with a single **syntax error** (a mistake in the way programming elements are strung together) will not work at all.



Programming Concepts

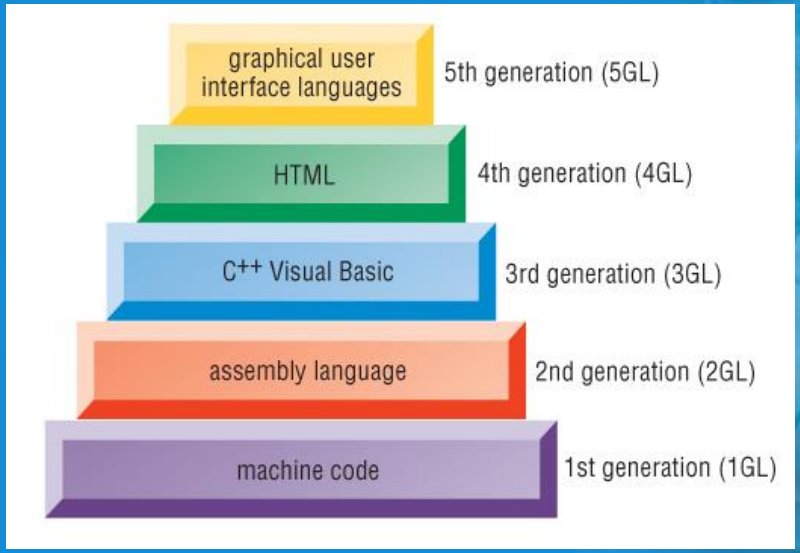
High-Level vs. Low Level Languages

- A **low-level language** is a binary language consisting of 1s and 0s.
  - *It is also known as **machine code**.*
  - *It runs faster and takes up less disk space.*
- A **high-level language** is relatively similar to natural languages such as English.
  - *It is easier to learn and use.*



Programming Concepts

- A computer **programming language generation** is a group of languages that were developed at the same time.
- Each generation builds on the contributions of the one before it.



Programming Concepts

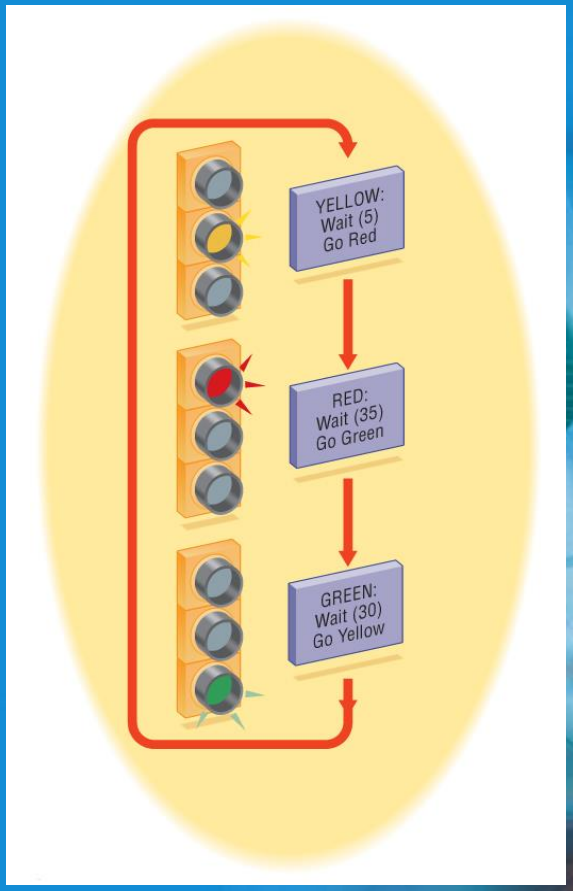
Classic Programming Elements

– The four main programming elements are:

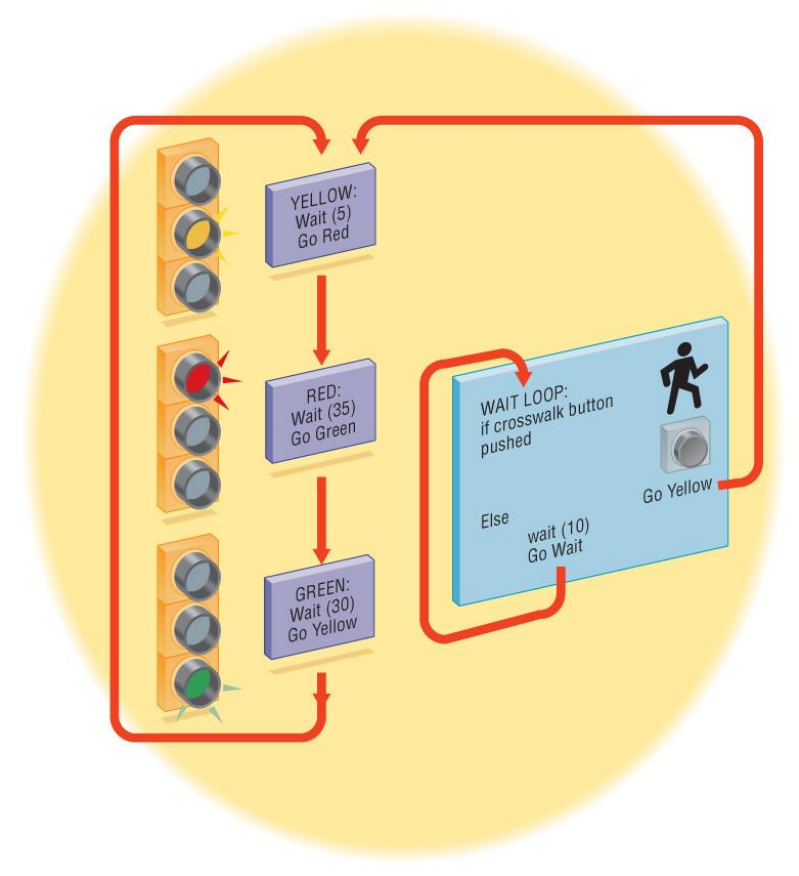
- ***variables*** *– data values stored in computer memory*
- ***executable statements*** *– perform actions and proceed to the next statement in the sequence*
- ***looping*** *– allows a program to return to a previously executed instruction and repeat it*
- ***decision statements*** *– points in a program where different actions may be performed depending on specific conditions*

Programming Concepts

Looping allows a computer program to continuously repeat the same steps, such as a program designed to direct a traffic light to display yellow, red, and green lights at a consistent rate.



Programming Concepts



A Decision Statement's Effect on Program Looping

Using an **if-then statement**, based on a particular action, such as pushing the crosswalk button, a program can interrupt the looping pattern, making the program more useful.

Problem-Solving Techniques

In the divide-and-conquer approach, programmers tackle one small piece of the puzzle at a time.

The **top-down design** approach helps programmers break a large project into manageable parts.

Problem-Solving Techniques

Problem-Solving Steps

1. Identify the problem.
2. Analyze the problem
3. Brainstorm solutions and choose the best one.
4. Write the algorithm.
5. Prototype the solution.
6. Implement and test the solution.

Problem-Solving Techniques

- An **algorithm** is a complete list of steps for solving a problem.
- Algorithms are written in **pseudocode**.
- The step-by-step pseudocode algorithm for changing a lightbulb appears at left.

The Evolution of Programming Approaches

Structured Programming

- This type of programming presents guidelines for an organized, logical approach to programming.
- The programmer thinks in terms of structured groups of instructions built on **routines**, which are sections of programs that handle specific functions.

Routines are then broken down into steps.

The Evolution of Programming Approaches

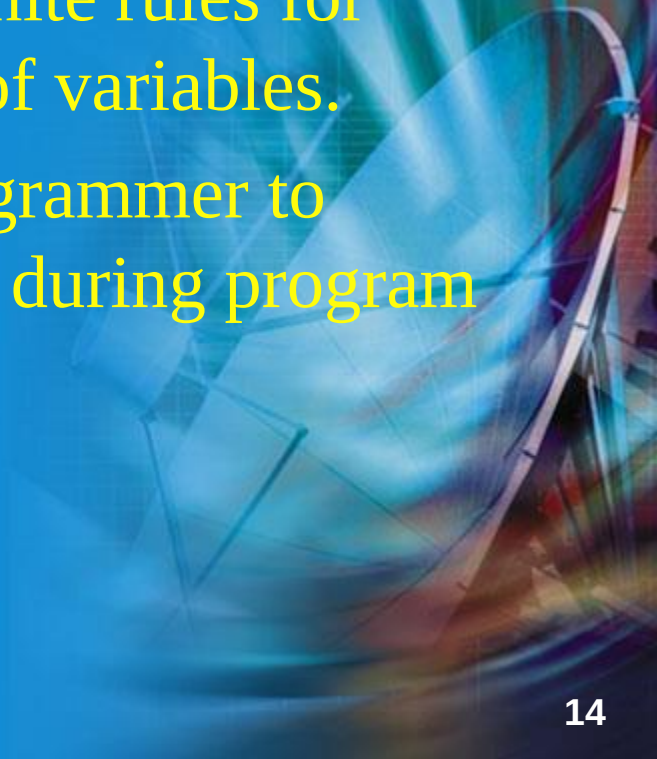
Modules

- In **modular code**, programmers create code modules that handle the separate components of a program.
  - *Modules are reusable and help in tracking down the source of errors.*
  - ***Modularity*** *describes how well source code is divided into individual modules.*
- A **macro** is a recording of steps to perform a repetitive activity.

The Evolution of Programming Approaches

Object-Oriented Programming

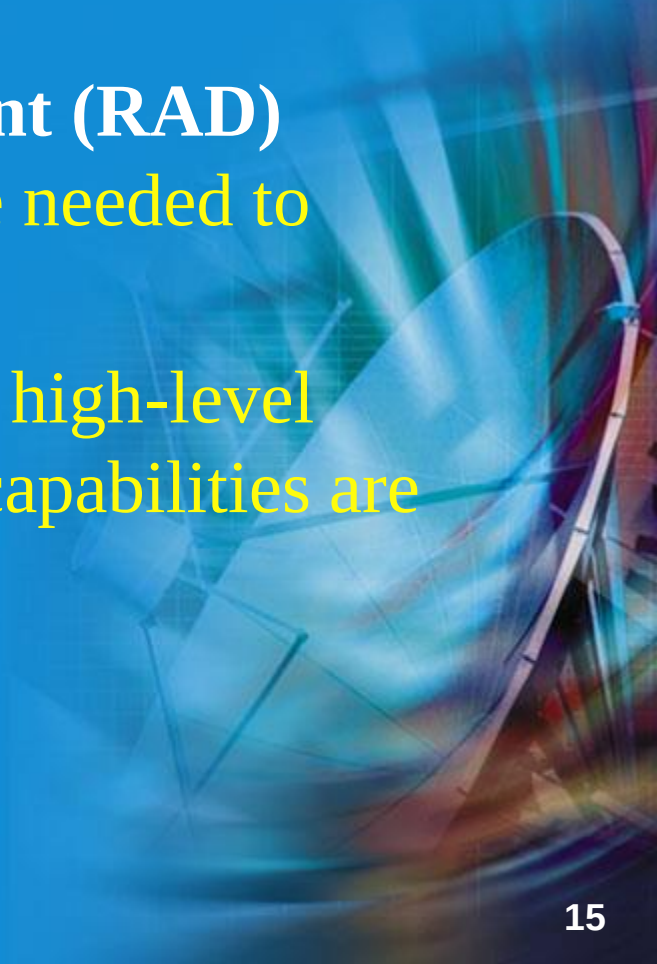
- **Object-oriented programming (OOP)** defines each module (**object**) with definite rules for interfacing and a protected set of variables.
- Protected variables allow a programmer to prevent data from being altered during program execution.



The Evolution of Programming Approaches

Rapid Application Development

- **Rapid application development (RAD)** reduces cost by decreasing time needed to develop a project.
- Visual Basic, Delphi, and other high-level languages with good interface capabilities are often used to aid in RAD.



The Evolution of Programming Approaches

Rapid Application Development

Programmers using RAD follow these guidelines:

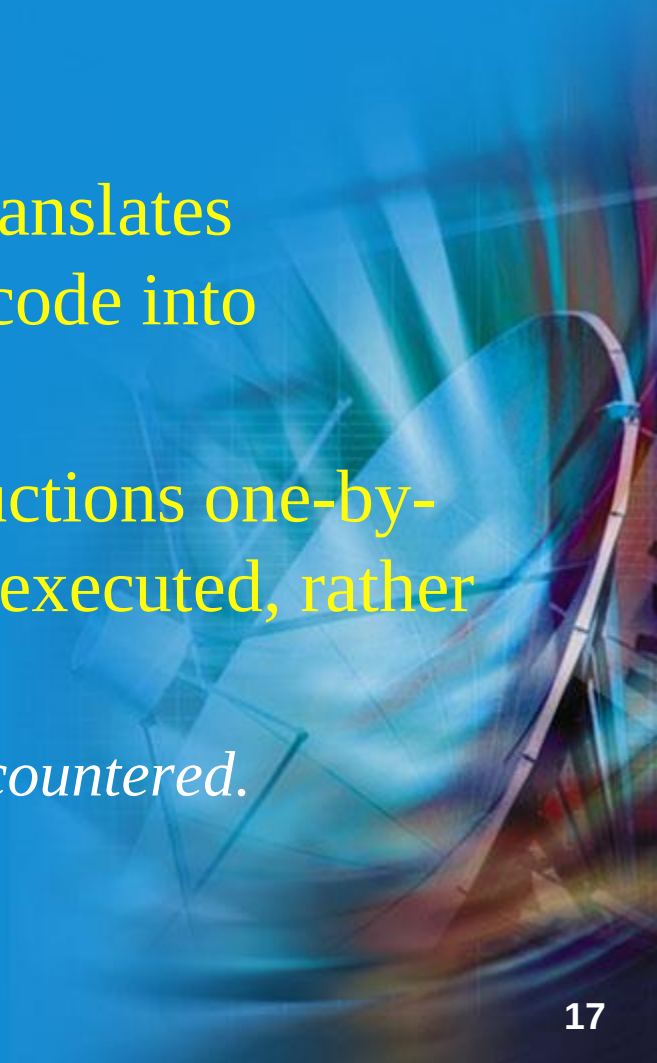
- *Use visual development tools whenever possible.*
- *Rapidly prototype new projects in order to reduce redesign time.*
- *Approach coding with these priorities:*
  - Use existing code first.
  - Buy someone else's existing code second.
  - Write new code last.

Programming Development and Documentation Tools

Compilers and Interpreters

- A **compiler** is a program that translates programming language source code into machine code.
- An **interpreter** translates instructions one-by-one as the source code is being executed, rather than all at once.

  It identifies errors as they are encountered.



Programming Development and Documentation Tools

Debuggers

- A **bug** is a computer error.
- A **debugger** is a software tool that helps programmers find errors quickly.
- Debuggers are an integral component of compilers and interpreters.

Programming Development and Documentation Tools

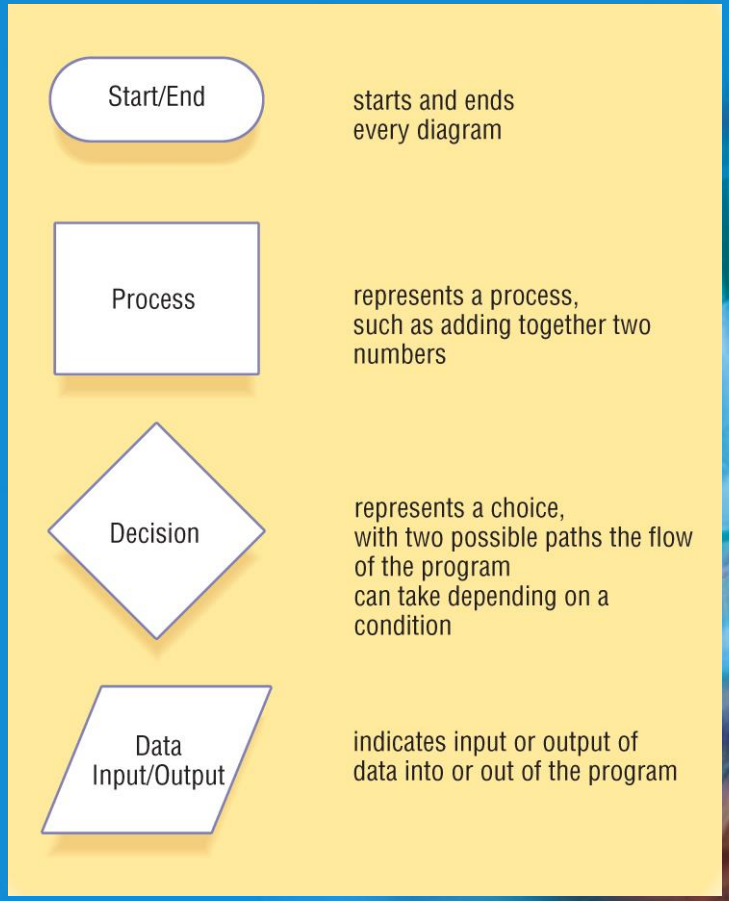
Documentation Tools

- The written notes that explain how a program works are called **documentation**.
  - *A **flowchart** provides a visual diagram of an algorithm.*
  - ***CASE tools** help a programming team schedule and coordinate its operations.*
  - *A **comment** is an informational message inserted into program source code to explain it to later readers.*

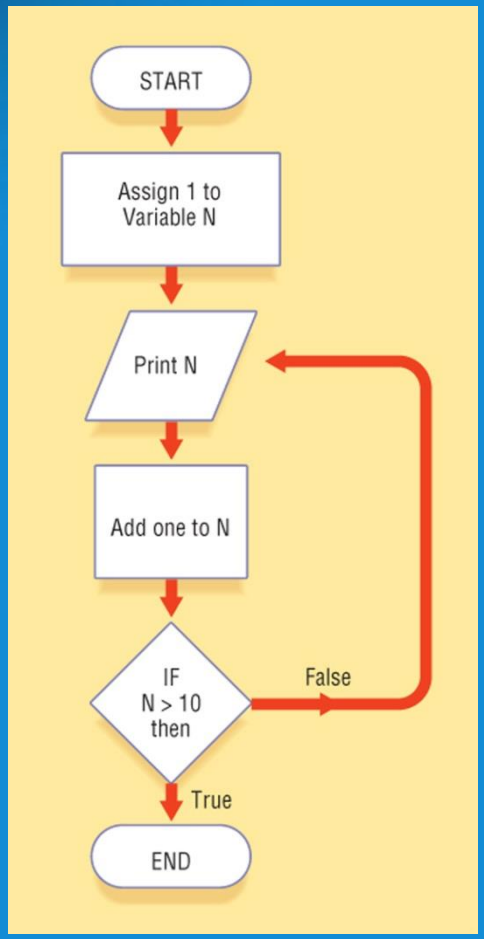
Programming Development and Documentation Tools

Flowchart Symbols

These symbols are used in flowcharts to represent the logic of a program.



Programming Development and Documentation Tools



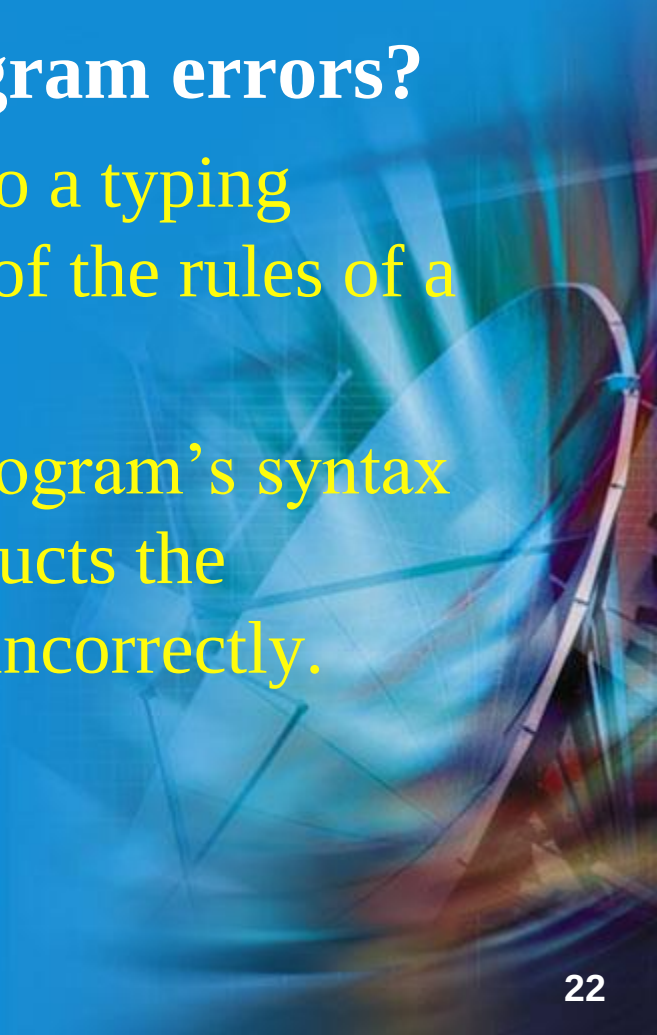
Algorithm Flowchart

Flowcharts help programmers visualize the steps in a software program.

Programming Errors

What are the main types of program errors?

- A **syntax error** is usually due to a typing mistake or a misunderstanding of the rules of a language.
- A **logic error** occurs when a program's syntax is correct, but the program instructs the computer to perform an action incorrectly.



Programming Errors

What are the main types of program errors?

- A **run-time error** refers to mistakes that occur when the application is running.
  - *A **crash bug** causes a program to stop running.*
  - *An **infinite loop** causes a program to perform the same set of instructions over and over.*
- A **style error** occurs as a result of poorly written programming code.

  ***Dead code** makes source code hard to read.*

Commonly Used Programming Languages

Language	Compiled	Interpreted	Object-Oriented
C	Yes	No	No
C++	Yes	No	Yes
Java	Yes	Yes	Yes
Visual Basic	No	Yes	No

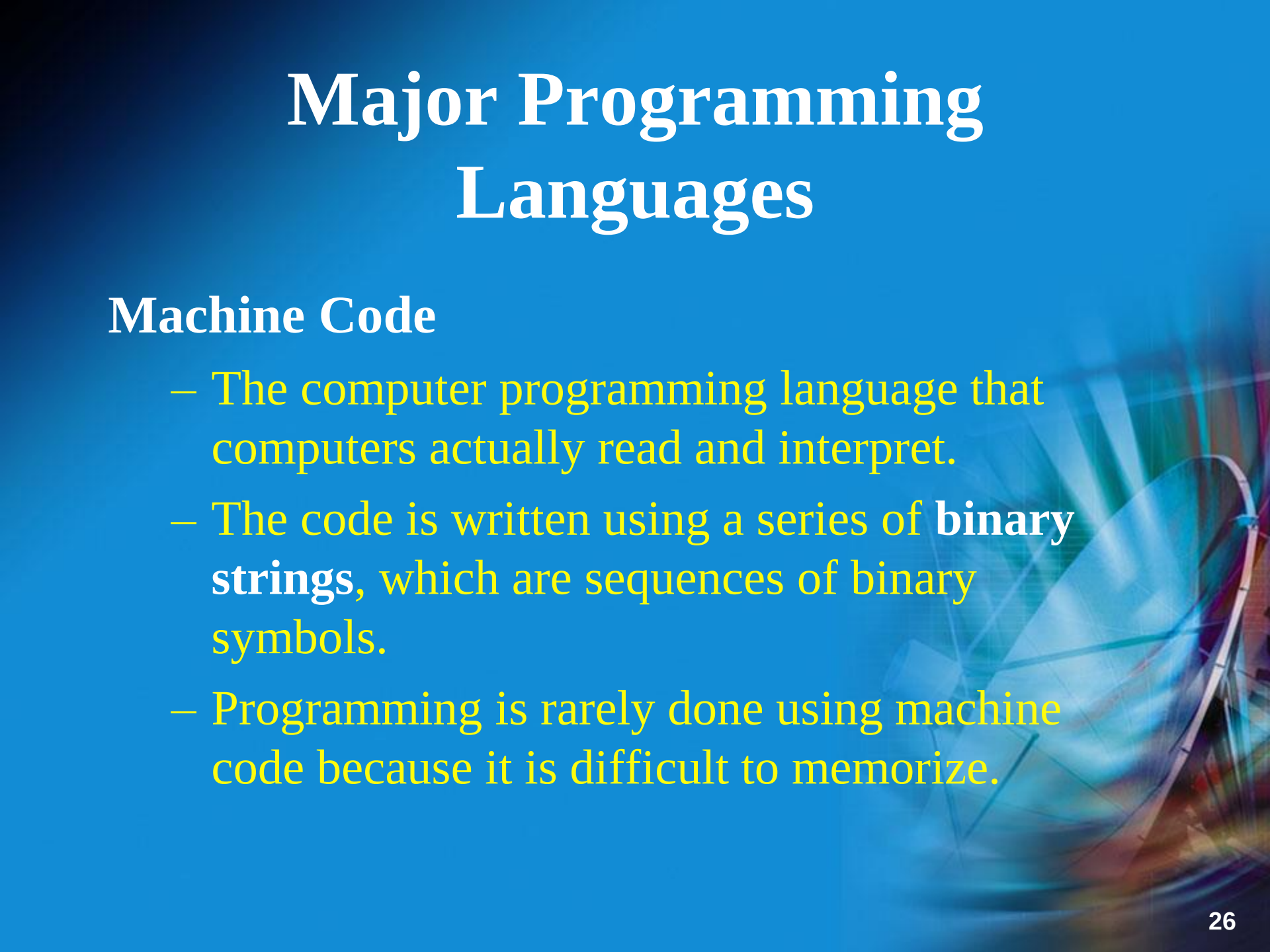
Interpreter and Compiler

Interpreter	Compiler
Translates program one statement at a time.	Scans the entire program and translates it as a whole into machine code.
It takes less amount of time to analyze the source code but the overall execution time is slower.	It takes large amount of time to analyze the source code but the overall execution time is comparatively faster.
No intermediate object code is generated, hence are memory efficient.	Generates intermediate object code which further requires linking, hence requires more memory.
Continues translating the program until the first error is met, in which case it stops. Hence debugging is easy.	It generates the error message only after scanning the whole program. Hence debugging is comparatively hard.
Programming language like Python, Ruby use interpreters.	Programming language like C, C++ use compilers.

Major Programming Languages

Machine Code

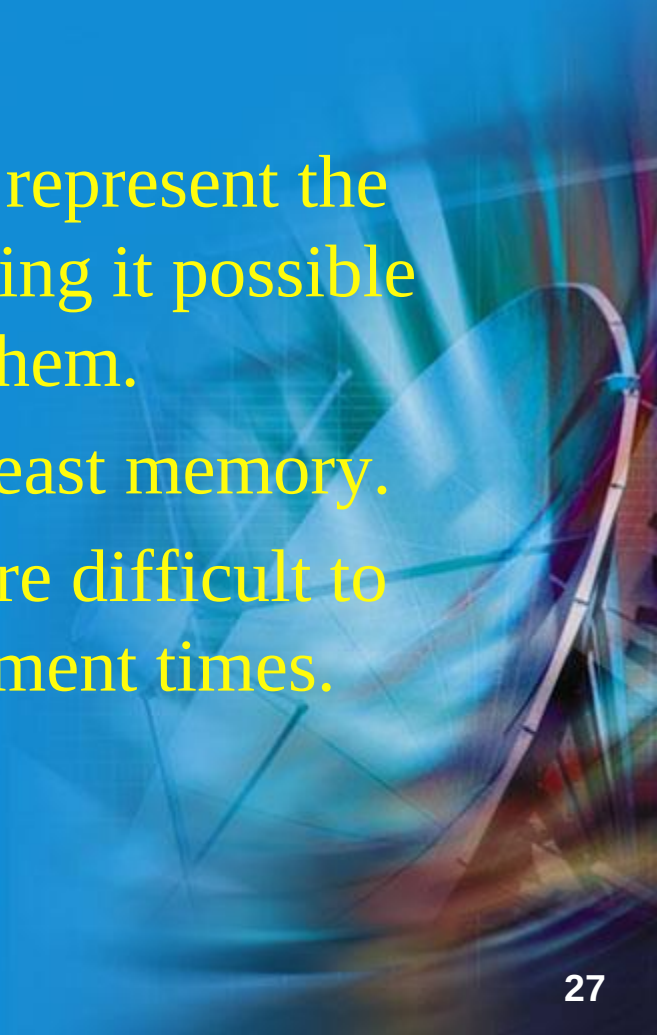
- The computer programming language that computers actually read and interpret.
- The code is written using a series of **binary strings**, which are sequences of binary symbols.
- Programming is rarely done using machine code because it is difficult to memorize.



Major Programming Languages

Assembly Language

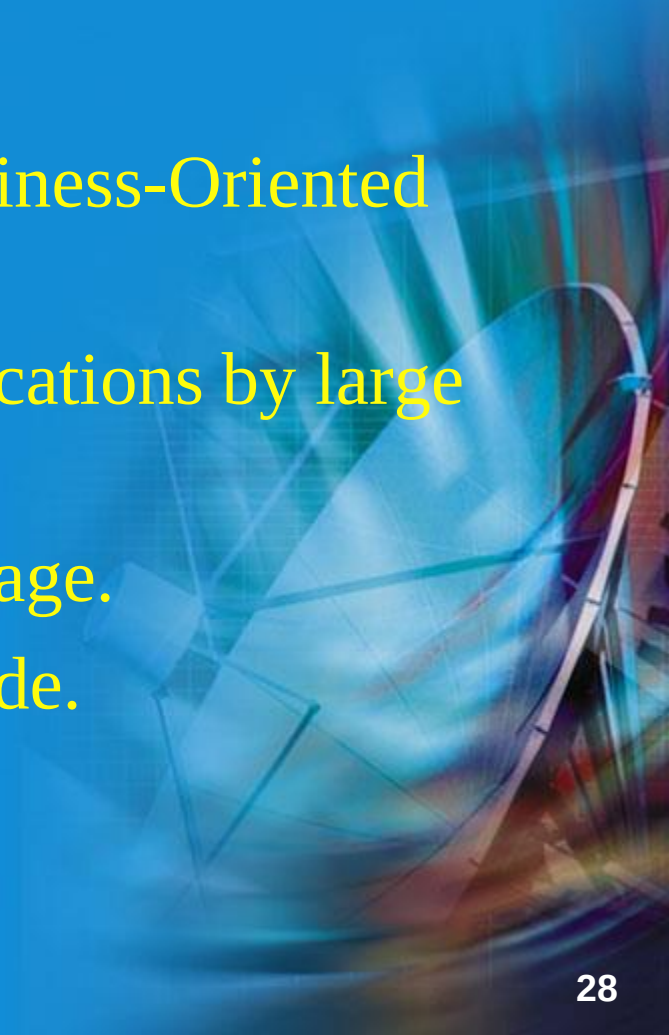
- Symbols and words are used to represent the elements of machine code, making it possible for programmers to memorize them.
- It runs the fastest and uses the least memory.
- Assembly language programs are difficult to write, which lengthens development times.



Major Programming Languages

COBOL

- An acronym for COmmon Business-Oriented Language.
- Used chiefly for business applications by large companies.
- A slow and cumbersome language.
- It has large body of existing code.



Major Programming Languages

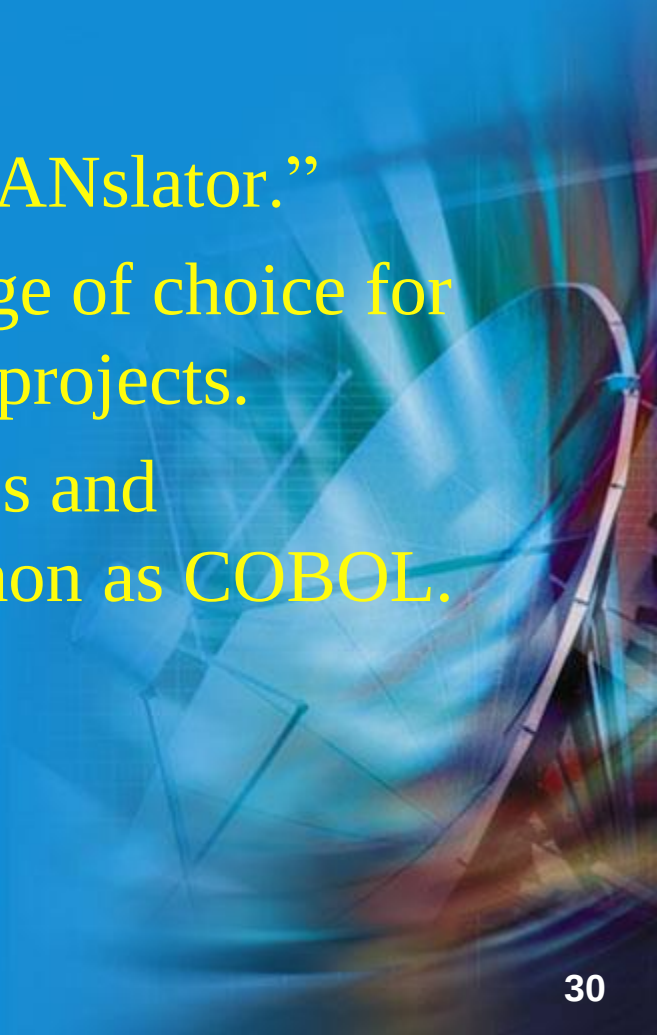
RPG

- An acronym for Report Program Generator.
- Commonly used in business environments.
- It is often used on midrange and mainframe computers.
- It is slow and inefficient, but simplifies the coding of database applications.
- It is familiar to many programmers.

Major Programming Languages

FORTRAN

- An acronym for "FORmula TRANslator."
- Was for many years the language of choice for math, science, and engineering projects.
- It is still in use today in factories and laboratories, but is not as common as COBOL.



Major Programming Languages

BASIC

- An acronym for "Beginner's All-purpose Symbolic Instruction Code."
- A high-level language.
- It is more natural than COBOL and FORTRAN.
- Today, BASIC is used professionally in an updated form, often Visual Basic.
- It runs slowly, but is faster to develop.

Major Programming Languages

Example of BASIC Source Code

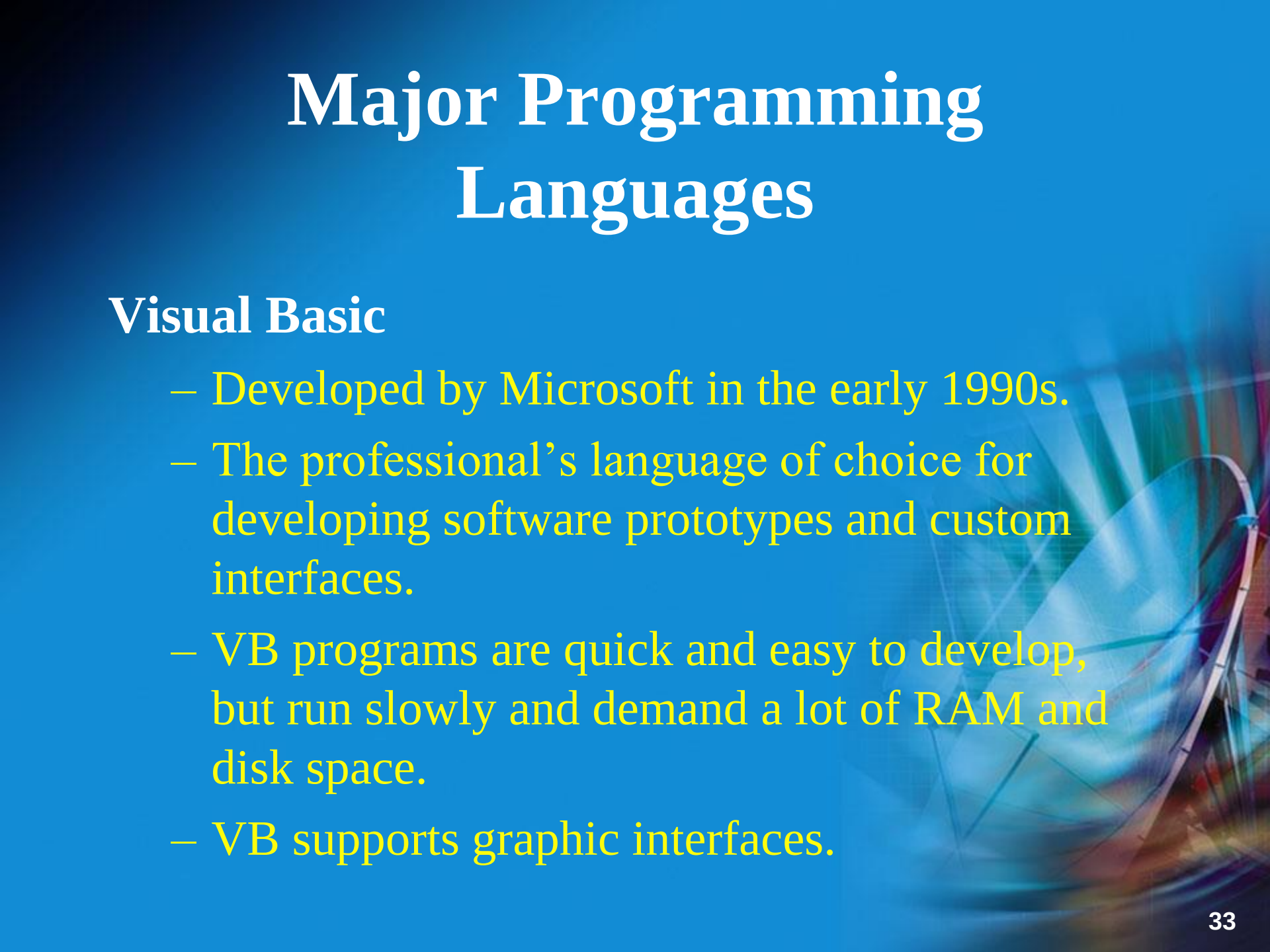
This sequence of code will print the numbers 1 to 10.

```
10  Rem This code sample prints out 1 through 10 on the screen.
20  For Count = 1 to 10
30  Print Count
40  Next Count
50  End
```

Major Programming Languages

Visual Basic

- Developed by Microsoft in the early 1990s.
- The professional's language of choice for developing software prototypes and custom interfaces.
- VB programs are quick and easy to develop, but run slowly and demand a lot of RAM and disk space.
- VB supports graphic interfaces.



Major Programming Languages

C

- Originally developed for the UNIX operating system.
- Named because it builds on an earlier language called “B.”
- C is a compromise between high- and low-level languages, but is considered a high-level language.
- C programs aren’t as easy to read as BASIC, but they run faster and use less disk space.

Major Programming Languages

C++

- A superset of C, with added features such as object-oriented programming.
- Any C program should run without problems as a C++ program.
- Most professional software sold today is written in the form of C or C++.

Major Programming Languages

C#

- Pronounced “See-Sharp.”
- An object-oriented language derived from C++ and Java.
- It combines the productivity of Visual Basic with the power of C++.
- It allows use of features in the Microsoft.NET framework, C, and Microsoft’s Component Object Model (COM).

Major Programming Languages

Scripting Languages

- An interpreted language that is relatively easy to learn and use.

  *It is a **nonprocedural language**— it explains what the computer should do in English-like terms, but not precisely how to do it.*

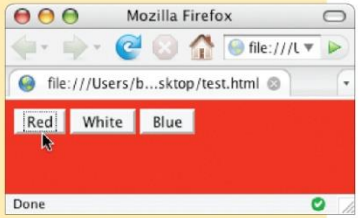
- **Hypertext Markup Language (HTML)**, **JavaScript**, **VBScript**, and **perl** are scripting languages.

Major Programming Languages

Example of JavaScript

This sequence of code directs a browser to display buttons that users can click to change the background color within the Web browser window.

```
<head>
<script type="text/javascript">

<!--Comment: This JavaScript Code will cause three buttons to appear that
can change the background color when clicked.-->

<!--Begin
function newcolor(aColor)
{
document.bgColor=aColor;
}
End-->
</script>

</head>

<body>

<div id="choices">
<form>
 <input type="button" value="Red" onclick="newcolor('red');">
 <input type="button" value="White" onclick="newcolor('white');">
 <input type="button" value="Blue" onclick="newcolor('blue');">
</form>
</div>

</body>
```



Any Question?