A. Tick the correct answers

1. How many generations can computer languages be classified into?
   c. Five
2. This generation of language uses mnemonic codes.
   a. Second
3. Which of the following is not an example of a high-level language?
   a. Machine
4. Which of the following does not translate the entire program at once?
   a. Interpreter
5. In which year were the compilers introduced?
   c. 1952

B. Fill in the blanks

1. **Machine language** is the only language that a computer understands.
2. Assembly language is regarded as the **second** generation language.
3. A program converted into machine language by a translator is called **object code**.
4. A/an **interpreter** converts a high-level language program into machine language, line-by-line.
5. **Prolog** (or OPS5, Mercury) is an example of fifth generation language.

C. Write T for True and F for False

1. A program is a set of instructions.
   T
2. The development of computer languages is classified into two categories.
   F (It's classified into five generations.)
3. Assembly language consists of binary numbers, 0s and 1s.
   F (It uses mnemonic codes like ADD, SUB.)
4. Machine language uses simple English words and phrases.
   F (Machine language uses 0s and 1s.)

5. The ultimate aim of 5GLs is to make computers behave, think, and react as human beings.
   T

D. Answer the following questions

1. What do you understand by a programming language? Define machine language.
A programming language is a set of specific instructions written to tell a computer how to perform tasks.
Machine language is the first-generation language consisting only of binary digits (0s and 1s). It is the only language that a computer's CPU understands directly.

2. What are Second Generation Languages? List the advantages and disadvantages of assembly language.
Second Generation Languages (2GL) are **assembly languages** that use mnemonic codes instead of long binary numbers.

- **Advantages:**
  - Easier to read and understand than machine language (uses English-like mnemonics).
  - Easier to debug and correct errors.
- **Disadvantages:**
  - Machine-dependent – different computers require different assembly languages.
  - Time-consuming to write programs.

3. Differentiate between source code and object code.

- **Source code** is the original code written by a programmer in assembly or high-level language.
- **Object code** is the translated version of source code in machine language, produced by a translator.

4. Define the term translator. How is an interpreter different from a compiler?
A **translator** (or language processor) converts source code (assembly or high-level) into machine code.

- **Compiler:** Translates the **entire** high-level program at once into machine code before execution.

- **Interpreter:** Translates and executes the program **line by line**, allowing quicker detection of errors but slower overall execution.

5. What are the characteristics of fifth-generation languages? Give some examples.
Characteristics:

- Used in **Artificial Intelligence (AI)** research.
- Eliminate the need for explicit algorithm writing.
- Programmer specifies **constraints and conditions**, and the computer finds solutions.
- Aim to make computers behave, think, and react like humans.

Examples: Prolog, OPS5, Mercury.

Extra Questions & Answers

Q1. What are Third Generation Languages (3GL)? Give examples.
Answer:
Third Generation Languages are **high-level languages** that are machine-independent, use English-like syntax and mathematical symbols, and are easier to understand.
Examples: FORTRAN, COBOL, BASIC, Pascal, C, C++, Java.

Q2. What are Fourth Generation Languages (4GL)? Mention their features.
Answer:
Fourth Generation Languages are **modern high-level languages** closer to human language and independent of operating systems.
Examples: SQL, Python, Ruby, Perl, PostScript.

Features:

- Very programmer-friendly and versatile.
- Require minimal coding effort.
- Use English-like syntax, icons, and symbolic representations.
- Faster execution than earlier generations.
- Reduce overall development time, effort, and cost.

Q3. What is the difference between an assembler and a compiler?
Answer:

- **Assembler:** Converts **assembly language** programs into machine language.
- **Compiler:** Converts **high-level language** programs into machine language in one go, translating the whole program after checking for syntax errors.

Q4. Why do computers need translators?
Answer:
Computers only understand machine language (binary). Translators (assemblers, compilers, interpreters) are needed to convert human-readable code into machine-readable binary code.