Language Fundamentals MCQs:

1. **Which of the following programming languages is often used for developing Android applications?**

   a) Java

   b) C#

   c) Swift

   d) Ruby

2. **Which programming paradigm encourages organizing code into reusable classes and objects?**

   a) Procedural Programming

   b) Functional Programming

   c) Object-Oriented Programming (OOP)

   d) Imperative Programming

3. **Which type of programming language represents instructions using binary code that can be directly executed by a computer's central processing unit (CPU)?**

   a) High-level languages
   b) Assembly languages
   c) Machine languages
   d) Scripting languages

4. **Which type of programming is often associated with more control over memory, registers, and hardware resources?**

   a) High-level programming
   b) Low-level programming
   c) Object-oriented programming
   d) Functional programming

5. **Assembly language instructions are converted into machine code by a program called:**
   a) Compiler
   b) Assembler
   c) Interpreter
   d) Linker

6. **High-level programming languages are mainly designed to:**
   a) Provide direct hardware access
   b) Be machine-dependent
   c) Be platform-independent
   d) Use only binary code

7. **High-level programming languages are designed to be:**
   a) Difficult to read and write
   b) Efficient for low-level tasks
   c) Easier for humans to understand and use
   d) Dependent on specific hardware architectures

8. **Which OOP principle allows a class to inherit properties parent classes?**
   a) Encapsulation
   b) Polymorphism
   c) Inheritance
   d) Composition

9. **Which of the following is NOT a core concept of the object-oriented programming paradigm?**

   a) Inheritance
   b) Abstraction
   c) Concurrency
   d) Polymorphism

10. **Which OOP principle allows a class to hide its internal details and provide a well-defined functionality?**

   a) Polymorphism
   b) Encapsulation
   c) Inheritance
   d) Abstraction

11. Which OOP principle is used to achieve different forms/implementation with the same name?

a) Polymorphism

b) Encapsulation

c) Inheritance

d) Abstraction

12. Which of the following is an advantage of using a compiler over an interpreter?

a) Faster execution

b) Easier debugging

c) More platform independence

d) Real-time interaction with code

13. Which of the following best describes an interpreter?

a) A program that translates source code into machine code

b) It executes the given code line by line

c) A tool for debugging programs

d) A program that converts high-level languages into assembly language

14. What is the main difference between a compiler and an interpreter?

a) Compilers execute code line by line, while interpreters convert code to machine language.

b) Compilers convert the entire source code into machine code at once, while interpreters execute code line by line.

c) Compilers and interpreters perform the same function but use different names.

d) Compilers translate machine code to source code, while interpreters execute source code directly.

15. Which of the following programming languages is typically associated with the procedural programming paradigm?

a) Java

b) Python

c) C

d) Ruby

16. In procedural programming, programs are organized around:

a) Objects and classes

b) Functions or procedures

c) Data structures

d) Inheritance hierarchies

17. Which programming paradigm emphasizes the concept of objects and their interactions?

a) Imperative programming
b) Object-oriented programming
c) Functional programming
d) Procedural programming

18. In object-oriented programming, what is the purpose of abstraction?

a) To define classes and objects
b) To represent data and its relationships
c) To hide complex implementation details and show only necessary features
d) To create instances of classes