2024-25

Roll No. ..... Total Pages : 3

43138

BT-3/D-24

PRINCIPLES OF PROGRAMMING LANGUAGES

Paper : ES227/205-A

Time : Three Hours] [Maximum Marks : 75

Note : Attempt *five* questions in all selecting at least *one* question from each Unit.

UNIT–I

1. (a) How do readability, writability, and reliability contribute to the quality of a programming language? (5)

   (b) How does a compiler convert high-level code to machine code? Explain. (5)

   (c) How do different programming languages handle the declaration of constants? (5)

2. (a) What is an enumeration type, and how does it improve code readability and error prevention? (5)

   (b) Explain the difference between implicit and explicit conversions. (5)

   (c) Explain what an attribute grammar is and how it extends context-free grammars. (5)

UNIT–II

3. (a) Explain how custom types are defined in a programming language, and discuss their benefits. (5)

   (b) What are the differences between static and dynamic type checking for structured data types? (5)

   (c) How are strings represented in programming languages? Explain any two operations. (5)

4. (a) How do unions impact memory usage, and what are some potential pitfalls when using them? (5)

   (b) What is subprogram overloading, and how is it supported in programming languages? (5)

   (c) How do ADTs improve program design and maintainability? (5)

UNIT–III

5. (a) What are deadlocks, and how do they relate to sequence control in concurrent subprograms?

   (b) How do exceptions differ from regular control structures in terms of sequence control?

   (c) Explain the difference between static and dynamic scoping.

6. (a) How does the referencing environment affect variable binding and lookup? (5)

   (b) Explain how recursion is implemented and managed within the call stack. (5)

(c) How does sequence control differ between single statements, compound statements, and blocks? (5)

UNIT–IV

7. (a) What are the primary runtime elements that require storage allocation during program execution? (5)

   (b) How does garbage collection relate to system-controlled storage management? (5)

   (c) How does stack-based storage management work, and what is its role in function calls? (5)

8. (a) Explain the principles of structured programming and how it improves code readability. (5)

   (b) Compare memory management in C and C++. (5)

   (c) What is functional programming, and how does it differ from object-oriented programming? (5)