S.E. (Computer) (Semester – III) (Revised 2007-08)
Examination, Nov./Dec. 2009
PRINCIPLES OF PROGRAMMING LANGUAGES

Duration : 3 Hours   Total Marks : 100

Instructions : *1) Attempt **any five** questions by selecting at least **one** from each*
Module.
*2) Assume suitable data **if necessary**.*

Module – I

1. a) What does the term "virtual machine" mean ? What are its advantages and
disadvantages ?   4

   b) Construct a parse tree for the following using BNF grammar :

      $P = ((Q * R) + (S * T))$

      Also comment on ambiguous and unambiguous grammar with examples.   8

   c) Explain type conversion and coercion with an example.   8

2. a) Draw a neat structure of a compiler and explain the following :   7

      i) Lexical Analysis

      ii) Syntactic Analysis

      iii) Semantic Analysis.

   b) Draw the NFA for the following :   6

      i) $(11*)(110 \ V \ 001)$

      ii) $(a + b) * ab(a + b)*$

   c) With the help of code illustrations using an object-oriented language provide
pros and cons of the following :   7

      i) Abstraction

      ii) Structured data types

P.T.O.

Module – II

3. a) Compare the execution for simple call-return subprograms with recursive subprograms. Draw necessary diagrams.   9

   b) Explain various control statements provided in C++.   5

   c) What is unification ? Which programming language supports it and how ?   6

4. a) Discuss pattern matching and backtracking with respect to sequence control of non-arithmetic expressions.   8

   b) Explicit return by a programmer or system has problems of garbage and dangling references. Explain with an example how this can be handled ?   6

   c) What do you understand by static scope and dynamic scope ?   6

Module – III

5. a) Explain the features with examples of block structured languages.   8

   b) Write short notes on the following :   12
      i) Semaphores
      ii) Monitors
      iii) Exception handling and issues.

6. a) Explain the significance of message passing with the help of an example.   6

   b) Explain giving illustration how the control is transferred between the coroutines. 8

   c) Why modules are important in large programming environment ? Give reasons with examples.   6

Module – IV

7. a) Why is LISP called as a Functional programming language ? Illustrate the features of using LISP syntax to support your answer.   8

   b) What do you understand by rules in Prolog ? How is a compound goal executed ? Explain how backtracking works to prove a compound goal in Prolog. Give an example.   8

   c) Differentiate between procedural language and object oriented language.   4

8. a) How does string handling differ in the following :   9

   i) C

   ii) C++

   iii) ADA

   Give comparisons using code examples.

   b) What is the importance of the following in Smalltalk :   6

   i) Data encapsulation

   ii) Inheritance.

   c) Differentiate between Procedural languages and Object Oriented languages.   5