

**III B. Tech I Semester Regular/Supplementary Examinations, October/November - 2017**  
**PRINCIPLES OF PROGRAMMING LANGUAGES**

(Computer Science and Engineering)

Time: 3 hours

Max. Marks: 70

- Note: 1. Question Paper consists of two parts (**Part-A** and **Part-B**)  
2. Answering the question in **Part-A** is compulsory  
3. Answer any **THREE** Questions from **Part-B**
- ~~~~~

**PART -A**

- 1 a) What are the factors influencing the writability of a language? [4M]  
b) List the advantages of using control structures in any of the compiled programming languages. [3M]  
c) Define Shallow and Deep binding for referencing environment of subprograms that have been passed as parameters. [4M]  
d) Describe briefly about Monitors. [4M]  
e) Write about Meta Language declaration statements. [4M]  
f) What is the relationship between resolution and unification in Prolog? [3M]

**PART -B**

- 2 a) Compare and contrast between the special purpose and general purpose programming languages. [4M]  
b) What is attribute grammar? Give the syntax directed definition for a desktop calculator. [8M]  
c) What are the limitations of recursive descent parser? [4M]
- 3 a) Explain the conditional statements and its implementation with examples. [8M]  
b) Explain the scope and lifetime of variables. Illustrate when they would coincide and when they don't. [8M]
- 4 a) Define a subprogram. Write the semantics of call and return of a subprogram. [8M]  
b) Discuss about nested subprograms with examples. [8M]
- 5 a) How message passing is implemented in Ada? Explain with examples. [8M]  
b) What is an event? How the events are handled in various OOP languages. [8M]
- 6 a) Discuss the fundamental concepts of lambda calculus. [8M]  
b) Explain about LISP functional programming language. [8M]
- 7 a) Discuss about basic elements of Prolog. [8M]  
b) Explain different types of propositions present in logic programming. [8M]

\*\*\*\*\*

**III B. Tech I Semester Regular/Supplementary Examinations, October/November - 2017****PRINCIPLES OF PROGRAMMING LANGUAGES**

(Computer Science and Engineering)

Time: 3 hours

Max. Marks: 70

- 
- Note: 1. Question Paper consists of two parts (**Part-A** and **Part-B**)  
 2. Answering the question in **Part-A** is compulsory  
 3. Answer any **THREE** Questions from **Part-B**
- ~~~~~

**PART -A**

- 1 a) Describe the approach of using axiomatic semantics to convert the correctness of a given program? [4M]  
 b) List the advantages and disadvantages of mixed mode arithmetic expressions. [4M]  
 c) Why is type checking the parameters of a subprogram important? [3M]  
 d) What is the primary problem with semaphores to provide synchronization? [4M]  
 e) Write a short note on ML functions. [4M]  
 f) What are the syntactic form and usage of fact and ruled statements in Prolog? [3M]

**PART -B**

- 2 a) How do you describe the meanings of programs using dynamic semantics? [4M]  
 b) Explain in detail about recursive descent parsing. [8M]  
 c) Give an example of left recursive rule in CFG. What is the significance of left Recursive rule? [4M]
- 3 a) Explain about the following [8M]  
 i) associative arrays ii) union types  
 b) State whether static binding is more reliable or dynamic binding. Justify. [8M]
- 4 a) Define a function. What are the design issues for functions? Explain. [8M]  
 b) Explain how subprogram is overloaded? Give examples. [8M]
- 5 a) Compare and contrast the cooperation synchronization and competition synchronization in message passing. [8M]  
 b) Explain the basic concepts of exception handling. [8M]
- 6 a) How ML is different from other functional programming languages? [8M]  
 b) Why were imperative features added to most dialects of LISP? [8M]
- 7 a) Explain how RDBMS and expert systems are helped using logic programming. [8M]  
 b) Discuss Terms and Goal statements in Prolog with examples. [8M]

\*\*\*\*\*

**III B. Tech I Semester Regular/Supplementary Examinations, October/November - 2017****PRINCIPLES OF PROGRAMMING LANGUAGES**

(Computer Science and Engineering)

Time: 3 hours

Max. Marks: 70

- Note: 1. Question Paper consists of two parts (**Part-A** and **Part-B**)  
 2. Answering the question in **Part-A** is compulsory  
 3. Answer any **THREE** Questions from **Part-B**

**PART -A**

- |   |  |      |
|---|--|------|
| 1 | a) Define grammar, derivation and a parse tree.                  | [4M] |
|   | b) What are the design issues for string types?                  | [3M] |
|   | c) What are generic methods?                                     | [4M] |
|   | d) List out the errors that can occur in expression evaluation.  | [4M] |
|   | e) What is type inferencing used in ML?                          | [4M] |
|   | f) Mention the various applications of multi paradigm languages. | [3M] |

**PART -B**

- |   |  |      |
|---|--|------|
| 2 | a) Explain language evaluation criteria and the characteristics that affect them.  | [8M] |
|   | b) Discuss the general approaches for the implementation of a Lexical analyzer.  | [8M] |
| 3 | a) Explain in detail arrays, indices, subscript bindings, and array categories.  | [8M] |
|   | b) Define unconditional branching. What are the problems with unconditional branching?                                     | [4M] |
|   | c) Discuss various methods for reclaiming garbage.   | [4M] |
| 4 | a) Discuss the design issues of subprogram and its operations performed on them.   | [8M] |
|   | b) Explain how subprogram names are passed as parameters.  | [8M] |
| 5 | a) Define a Thread. How are threads different from processes? Explain java threads with examples.                          | [8M] |
|   | b) Define monitor. Explain how cooperation synchronization and competition synchronization are implemented using monitors. | [8M] |
| 6 | a) Explain about scheme functional programming language.   | [8M] |
|   | b) Discuss how Haskell differs from ML.  | [8M] |
| 7 | a) Correlate the importance of logic programming languages over functional programming languages.                          | [8M] |
|   | b) Explain Fact and Rule Statements in Prolog with suitable examples.  | [8M] |

\*\*\*\*\*



**III B. Tech I Semester Regular/Supplementary Examinations, October/November - 2017****PRINCIPLES OF PROGRAMMING LANGUAGES**

(Computer Science and Engineering)

Time: 3 hours

Max. Marks: 70

---

Note: 1. Question Paper consists of two parts (**Part-A** and **Part-B**)

2. Answering the question in **Part-A** is compulsory

3. Answer any **THREE** Questions from **Part-B**

---

**PART -A**

- 1 a) What are the difficulties in using an attribute grammar to describe all of the syntax and static semantics of a contemporary programming language? [4M]
- b) Write a note on Boolean and relational expressions. [3M]
- c) State the importance of Local Referencing Environments with suitable examples. [4M]
- d) Differentiate between physical and logical concurrency. [4M]
- e) What scoping rules are used in ML? [3M]
- f) Describe the multi - paradigm languages. [4M]

**PART -B**

- 2 a) Discuss in detail about the attribute grammars. [4M]
- b) Explain how is the order of evaluation of attributes determined for the tree of a given grammar. [8M]
- c) Why lexical and syntax analyzer are separated out? [4M]
- 3 a) Discuss the merits of guarded commands. [3M]
- b) What is a variable? What are the attributes of a variable? Elaborate on address of a variable. [8M]
- c) Explain in detail about overloaded operators. [5M]
- 4 a) Discuss how generic methods are implemented with suitable examples. [8M]
- b) Explain the importance of dynamic scoping with an example. [8M]
- 5 a) What are the three possible levels of concurrency in programs? Explain. [8M]
- b) Discuss the reasons for using exception handlers in a programming language. [8M]  
What if there exist programming languages with no exception handlers.
- 6 a) Give comparison of Functional and Imperative Languages. [8M]
- b) Explain the control structure of a PROLOG program. [8M]
- 7 a) How PROLOG is different from other logic programming languages? Give an example for each feature. [8M]
- b) Explain Prolog interfacing process. [8M]

\*\*\*\*\*

