Code No: R05320502 **Set No. 1**

III B.Tech II Semester Regular Examinations, Apr/May 2008
COMPILER DESIGN
(Computer Science & Engineering)

Time: 3 hours **Max Marks: 80**

Answer any FIVE Questions
All Questions carry equal marks

⋆ ⋆ ⋆ ⋆ ⋆

1. Explain the input buffer scheme for scanning the source program. How the use of sentinels can improve its performance? Describe in detail. [16]

2. Construct predictive parsing table for the following grammar.
   $E \rightarrow T\ E'$
   $E' \rightarrow +T\ E'|\varepsilon$
   $T \rightarrow F\ T'$ [16]
   $T' \rightarrow {}^*F\ T'|\varepsilon$
   $F \rightarrow (E)|id$

3. (a) What is an operator grammar? Give an example.
   (b) Write an operator precedence parsing algorithm. [6+10]

4. (a) Write a note on the specification of a simple type checker.
   (b) What is a type expression? Explain the equivalence of type expressions with an appropriate examples. [8+8]

5. (a) Compare three different storage allocation strategies.
   (b) Consider the following array declaration in 'c';
   float a[100][100];
   Assume that the main memory in byte addressable and that the array is stored starting from the memory address 100. What is the address of a[40][50]?[8+8]

6. Explain different principal sources of optimization technique with suitable examples. [16]

7. (a) Write and explain live variable analysis algorithm.
   (b) Explain the use of algebraic transformations with an example [8+8]

8. (a) Explain the different issues in the design of a code generator.
   (b) Generate code for the following C statements:
      i. x= f(a) + f(a) + f(a)
      ii. x= f(a) /g(b,c)
      iii. x= f(f(a))
      iv. x= ++f(a) [8+8]

⋆ ⋆ ⋆ ⋆ ⋆

Code No: R05320502 **Set No. 2**

III B.Tech II Semester Regular Examinations, Apr/May 2008
COMPILER DESIGN
(Computer Science & Engineering)

Time: 3 hours **Max Marks: 80**

Answer any FIVE Questions
All Questions carry equal marks

⋆ ⋆ ⋆ ⋆ ⋆

1. (a) Explain the different phases of a compiler, showing the output of each phase, using the example of the following statement:
position : = initial + rate * 60
   (b) Compare compiler and interpreter with suitable diagrams. [10+6]

2. (a) What is recursive descent parser? Construct recursive descent parser for the following grammar.
$E \rightarrow E + T|T$
$T \rightarrow TF|F$
$F \rightarrow F^*|a|b$
   (b) What is ambiguous grammar? Eliminate ambiguities for the grammar:
$E \rightarrow E + E|E^*E|(E)|id.$ [8+8]

3. Construct SLR parsing table for the following grammar.
$S \rightarrow AS|b$
$A \rightarrow SA|a$ [16]

4. (a) Write the quadruple, triple, indirect triple for the statement $a := b^* - c + b^* - c$.
   (b) Explain the role of intermediate code generator in compilation process. [8+8]

5. Write an algorithm to perform the table lookup and insertion operation for hashed symbol table. [16]

6. (a) What is code optimization? What are its advantages?
   (b) Explain briefly about folding.
   (c) What are the problems in optimizing compiler design? [5+5+6]

7. (a) Explain reducible and non-reducible flow graphs with an example.
   (b) Explain natural loops and inner loops of a flow graph with an example. [8+8]

8. (a) Explain the concept of object code forms.
   (b) Generate optimal machine code for the following C program. [6+10]

```
main()
{
   int i, a[10];
   while (i<=10) a[i] =0
}
```

⋆ ⋆ ⋆ ⋆ ⋆

Code No: R05320502 **Set No. 3**

III B.Tech II Semester Regular Examinations, Apr/May 2008
COMPILER DESIGN
(Computer Science & Engineering)

Time: 3 hours **Max Marks: 80**

Answer any FIVE Questions
All Questions carry equal marks

⋆ ⋆ ⋆ ⋆ ⋆

1. (a) Explain, in detail, lexical analyzer generator.
   (b) Describe the lexical errors and various error recovery strategies with suitable examples. [8+8]

2. (a) Consider the following grammar.
   S → 0A|1 B|0| 1
   A → 0S|1 B| 1
   B → 0 A|1 S
   Construct leftmost derivations and parse trees for the following sentences
   i. 0101
   ii. 1100101
   (b) Consider the following grammar
   E → T + E|T
   T → V*T|V
   V → id
   Write down the procedures for the nonterminals of the grammar to make a recursive descent parser. [8+8]

3. (a) Define LR(k) parser. Draw and explain model of LR parser.
   (b) Write LR parsing algorithm. [8+8]

4. (a) Write the quadruple, triple, indirect triple for the statement a := b* − c + b* − c.
   (b) Explain the role of intermediate code generator in compilation process. [8+8]

5. (a) What is an ordered and unordered symbol table? What is the function of symbol table in the compliation process? Explain.
   (b) What are the various attributes of a Symbol Table? [10+6]

6. Explain different principal sources of optimization technique with suitable examples. [16]

7. Explain about data flow analysis of structured programs. [16]

8. (a) Explain the concept of object code forms.

(b) Generate optimal machine code for the following C program. [6+10]

```
main()
{
   int i, a[10];
   while (i<=10) a[i] =0
}
```

⋆ ⋆ ⋆ ⋆ ⋆

Code No: R05320502 **Set No. 4**

III B.Tech II Semester Regular Examinations, Apr/May 2008
COMPILER DESIGN
(Computer Science & Engineering)

Time: 3 hours **Max Marks: 80**

Answer any FIVE Questions
All Questions carry equal marks

⋆ ⋆ ⋆ ⋆ ⋆

1. (a) Consider the following fragment of 'C' code:
   float i, j;
   i = i * 70 + j + 2;
   Write the output at all phases of the compiler for the above 'C' code.
   (b) Write short notes on: input buffering. [10+6]

2. (a) What is recursive descent parser? Construct recursive descent parser for the following grammar.
   $E \rightarrow E + T|T$
   $T \rightarrow TF|F$
   $F \rightarrow F^*|a|b$
   (b) What is ambiguous grammar? Eliminate ambiguities for the grammar:
   $E \rightarrow E + E|E^*E|(E)|id.$ [8+8]

3. Construct SLR parsing table for the following grammar.
   $S \rightarrow AS|b$
   $A \rightarrow SA|a$ [16]

4. Write short notes on the following:
   (a) S-attributed definitions.
   (b) L-attributed definitions.
   (c) Dependency graph. [6+6+4]

5. (a) What is an ordered and unordered symbol table? What is the function of symbol table in the compliation process? Explain.
   (b) What are the various attributes of a Symbol Table? [10+6]

6. Explain different principal sources of optimization technique with suitable examples. [16]

7. A flow graph is useful for understanding code generation algorithm? Justify your answer with an example. [16]

8. Discribe various Register allocation optimization techniques with an example. [16]

⋆ ⋆ ⋆ ⋆ ⋆