2 0 2 2

(May/June)

COMPUTER SCIENCE

(Honours)

(Compiler Design)

(CS–602 AT)

Marks : 75

Time : 3 hours

*The figures in the margin indicate full marks
for the questions*

Answer **one** question from each Unit

UNIT—I

1. *(a)* Briefly explain the first two phases of a compiler with the help of an example.   6

   (b) What is the use of regular expressions? Provide the rules that define regular expressions.   2+7=9

2. *(a)* Write down the regular expressions for the following :   2+2+2=6

   (i) Identifiers (an identifier can start with a letter followed by either letters or digits)

(ii) Binary strings such as a '0' is always followed by a '1'

(iii) To check correct syntax for the e-mail address

(b) What does the following regular expressions mean? 2+2=4

(i) a(ab)*a

(ii) [0–7][0–7]*

(c) What do you understand by finite automata? Differentiate between non-deterministic finite automata (NFA) and deterministic finite automata (DFA). 2+3=5

UNIT—II

3. *(a)* Differentiate between left derivation and right derivation. 4

(b) Write context free grammars to detect the following. Is your grammar ambiguous? 3+3=6

(i) Occurrence of balanced parentheses

(ii) Strings over the alphabet set {a, b}, such that every *a* is immediately followed by a *b*.

(c) Explain operator grammar and the precedence relationship symbols with the help of an example. 5

4. *(a)* Explain the concept of shift reduce parsing. 10

(b) What is left recursion? How can we eliminate it? 2+3=5

UNIT—III

5. *(a)* Compare and contrast static and dynamic checking. Give an example of a situation in which dynamic checking is really helpful. 4+2=6

(b) What is the associated information stored in a symbol table for an identifier? How is this information used during error detection? 5+4=9

6. *(a)* Why is type equivalence important during type checking? Draw the type trees for the following pairs of type declaration and argue whether they are type equivalent or not : 2+5=7

```
struct list 1 {          struct list 2 {
    Char x;                  char abc;
    int y;                   int xy;
    int a[100];              int ar[100];
}                        }
```

(b) Describe any two data structures that can be used to represent a symbol table. 8

UNIT—IV

7. *(a)* What do you mean by runtime storage allocation? Explain the difference between static and dynamic allocation. Why is it the case that statically allocated languages cannot support recursion? 2+4+2=8

(b) What is the purpose of intermediate code generation in overall compiler design? Describe in brief Abstract Syntax Trees and Directed Acyclic Graphs. 3+4=7

8. *(a)* What is an activation record? Briefly explain the components of an activation record. 2+5=7

(b) Show the annotated parse tree and code generation process for the following arithmetic expression : 3+5=8

$$a + (b - c) * d$$

UNIT—V

9. *(a)* Name and describe three factors that can affect code generation. 7

(b) Consider the following sequence of statements :

$$x = y + z$$
$$w = p + y$$
$$y = y * z$$
$$p = w - x$$

(i) Construct the corresponding DAG.

(ii) Perform code generation assuming two registers are available. 4+4=8

10. *(a)* Mention two factors that influence the optimization process. Specify the necessary and sufficient conditions for performing : 2+6=8

(i) Constant propagation

(ii) Dead code elimination

(iii) Loop optimization

(b) What is a basic block? Write down the intermediate code for the following code fragment and show the basic blocks and control flow within them : 2+5=7

```
begin
    prod = 1
    term = 1
    while term <=20 do
        prod = prod * term
        term = term +1
    end
end
```

★ ★ ★