## 2017/EVEN/08/24/MCS–802 (C)/ MS–202 (C)/CS–802 (O)/570

**PG Even Semester (CBCS) Exam., May—2017**

COMPUTER SCIENCE

**( 8th Semester/2nd Semester )**

Course No. : MCS–802 (C)/MS–202(C)/CS–802 (O)

**( Principles of Compiler Design )**

*Full Marks* : 75

*Pass Marks* : 30 (for CBCS)
26 (for Non-CBCS)

*Time* : 3 hours

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

Answer **five** questions, selecting **one** from each Unit

UNIT—I

**1.** *(a)* What are the rules that can be used to construct regular expressions over a certain alphabet Σ? 4

*(b)* What do the following regular expressions denote? 5

*(i)* $(a|b)^*$

*(ii)* $a|ba^*$

*(iii)* $(aa\,|\,ab\,|\,ba\,|\,bb)^*$





*(c)* What is a lexical analyzer? State the needs for lexical analysis. 6

**2.** *(a)* What is an NFA? How does it differ from a DFA? 4

*(b)* Draw a DFA to show all numbers divisible by 5 over the alphabet $\{0, 1\}$. 5

*(c)* Show, with a suitable example, how an NFA can be constructed from a regular expression. 6

UNIT—II

**3.** *(a)* What is a predictive parser? 3

*(b)* Distinguish between top-down parsing and bottom-up parsing with a suitable example. 5

*(c)* Is the following grammar LALR? Justify your answer : 7

$$S \to Aa\,|\,bAc\,|\,Bc\,|\,bBa$$
$$A \to d$$
$$B \to d$$

**4.** *(a)* What are LR parsers? Explain how LR(1) parsing can eliminate the deficiencies of SLR parsing. 2+4=6

*(b)* Is the following grammar SLR? If not, can the deficiency be rectified? Explain : 9

$$S \to Aa\,|\,bAc\,|\,dc\,|\,bda$$
$$A \to d$$

UNIT—III

**5.** *(a)* What is a symbol table? What are the information that can be associated with a name in the symbol table? 3

*(b)* Explain, with a neat diagram, the set of operations when the end of a block is reached. 5

*(c)* Give the output of the following program using pass-by value, pass-by reference, copy restore and pass-by name. Also give a brief explanation for the outputs : 7

```
int i;
int a [3];
void swap (int x, int y)
{
   x = x+y;
   y = x–y;
   x = x–y;
}
main ( )    {
   i = 1;
   a[0] = 2;
   a[1] = 1;
   a[2] = 0;
swap (i, a[i]);
printf ("%d %d %d %d|n", i, a[0],
   a[1], a[2]);
swap (a[i], a[i]);
printf ("%d %d %d|n", a[0], a[1], a[2]);
return 0; }
```





**6.** *(a)* What is a runtime environment? Explain the logical address space of a program with a neat diagram. 4

*(b)* What is stack-based allocation? Why is it the case that statically allocated languages cannot support recursion? 5

*(c)* Consider the following piece of code :

```
int x = 2;
void f (int n)   {
  static int x = 1;
  g (n);
  x – –;          }
void g (int m) {
int y = m–1;
if (y > 0)   {
f (y);
x – –;
             }
                }
main ( ) {g (x);
           return 0;
          }
```

Show the snapshot of activation records when f is called. Show the snapshot of activation records when g is called the second time. 3+3=6

## UNIT—IV

**7.** *(a)* What are the conditions that must be satisfied by the compiler while it attempts optimization? 3

*(b)* Expound the problems in designing an optimizing compiler. 5

*(c)* Explain the factors influencing code optimization. 7

**8.** Write short notes on the following : 5×3=15

*(a)* Loop interchange

*(b)* Loop fission

*(c)* Loop splitting

## UNIT—V

**9.** *(a)* Discuss the factors affecting code generation. 7

*(b)* Explain, with a suitable example, a register allocation scheme based on graph colouring. 8

**10.** *(a)* Explain peephole optimization. 7

*(b)* Discuss, with a suitable example, a code generation scheme using dynamic programming. 8

★ ★ ★