

C

Reg. No. :

--	--	--	--	--	--	--	--	--	--

Question Paper Code: 56201

B.E./B.Tech. DEGREE EXAMINATION, NOV 2019

Sixth Semester

Computer science and Engineering

15UCS601- PRINCIPLES OF COMPILER DESIGN

(Regulation 2015)

Duration: Three hours

Maximum: 100 Marks

PART A - (5 x 1 = 5 Marks)

1. A bottom up parser generates CO1-R
(a) Right most derivation (b) Rightmost derivation in reverse
(c) Leftmost derivation (d) Leftmost derivation in reverse
2. Which of the following is not a function of backpatching? CO2-R
(a) Backpatch(p,i) (b) Backpatch(i,p) (c) Makelist(i) (d) Merge(p1,p2)
3. Grammar that produce more than one Parse tree for same sentence is CO3-U
(a) Ambiguous (b) Unambiguous
(c) Complementary (d) Concatenation Intersection
4. Code optimizations are carried out on the intermediate code because CO4-U
(a) Program is more accurately analyzed on intermediate code than on machine code
(b) Optimization information from data flow analysis cannot be used
(c) They enhance the portability of the compiler to the other target processor
(d) Optimization information from the front end cannot be used
5. _____ reduces the dimensions of an array and reduces the number of memory locations accessed CO5-R
(a) Locality of Computed Data (b) Array contraction
(c) Pipelining (d) Communication Cost

PART – B (5 x 3= 15Marks)

6. Mention the advantages of having each phase of the compiler separately CO1-U
7. Construct a DAG for the expression: $a * (b - c) + (b - c) | d$ CO2-U

8. What is three-address code? CO3-R
9. Write about Constant folding. CO4-R
10. Point out the techniques for improving data locality in uniprocessors and multiprocessors CO5-R

PART – C (5 x 16= 80Marks)

11. (a) Examine the different phases of compiler. Translate the following statement into its equivalent target assembly code. $a := b + c * 60$ CO1-App (16)

Or

- (b) (i) Discover the purpose of compiler construction tools? explain it. CO1- R (8)
- (ii) Explain input buffering techniques with suitable code? CO1-R (8)

12. (a) Construct parse tree for the input string $w=id+id*id$ using top down parser. CO2-Ana (16)

$E \rightarrow TE'$

$E' \rightarrow +TE' \mid \epsilon$

$T \rightarrow FT'$

$T' \rightarrow *FT' \mid \epsilon$

$F \rightarrow (E) \mid id$

Or

- (b) Construct stack implementation of shift reduce parsing for the grammar CO2- Ana (16)

$E \rightarrow E+E$

$E \rightarrow E * E$

$E \rightarrow (E)$

$E \rightarrow id$ and the input string $id1+id2*id3$.

13. (a) Explain the semantic actions required for translation of array reference. CO3-U (16)

Or

- (b) Generate the intermediate code for the following code segment along with the required syntax directed translation scheme: CO3-Ana (16)

```
while(i<10)
  if(i%2==0)
    evensum=evensum+i;
  else
    oddsum=oddsum+i
```

14. (a) (i) Construct DAGs for the following basic blocks of TACs CO4 -Ana (8)

$$a = b + c ; \quad b = a - d ; \quad c = b + c ; \quad d = a - d ;$$
- (ii) Explain structure preserving transformation in detail using CO4-Ana (8)
basic blocks. $a = b + c ; \quad b = a - d ; \quad c = b + c ; \quad d = a - d ;$
- Or
- (b) Explain the ways in which the storage allocation is done in stack CO4- U (16)
heap memory management.
15. (a) (i) Explain the principal sources of optimization techniques with CO5-U (8)
suitable examples
- (ii) Write detailed notes on the peephole optimization techniques. CO5-U (8)
- Or
- (b) Explain about the Data reuse, its types in detail CO5-U (16)

