TE SPCC Quiz 1

Total points 10/10

Introduction to system softwares

The respondent's email address (**ameythakur@ternaengg.ac.in**) was recorded on submission of this form.

0 of 0 points

ROLL NO *

- ○ 1
- ○ 2
- ○ 3
- ○ 4
- ○ 5
- ○ 6
- ○ 7
- ○ 8
- ○ 9
- ○ 10
- ○ 11
- ○ 12
- ○ 13
- ○ 14
- ○ 15
- ○ 16
- ○ 17
- ○ 18
- ○ 19
- ○ 20
- ○ 21
- ○ 22
- ○ 23
- ○ 24

- 25
- 26
- 27
- 28
- 29
- 30
- 31
- 32
- 33
- 34
- 35
- 36
- 37
- 38
- 39
- 40
- 41
- 42
- 43
- 44
- 45
- 46
- 47
- 48
- 49
- [✓ 50]

- ○ 51
- ○ 52
- ○ 53
- ○ 54
- ○ 55
- ○ 56
- ○ 57
- ○ 58
- ○ 59
- ○ 60
- ○ 61
- ○ 62
- ○ 63
- ○ 64
- ○ 65
- ○ 66
- ○ 67
- ○ 68
- ○ 69
- ○ 70
- ○ 71
- ○ 72
- ○ 73
- ○ 74
- ○ 75
- ○ 76

- ○ 77
- ○ 78
- ○ 79

STUDENT NAME *

AMEY THAKUR

Introduction to System Softwares 10 of 10 points

✓ 1. Intermediate code generation phase gets input from 1/1

- ○ Lexical analyzer
- ○ Syntax analyzer
- [✓ Semantic analyzer] ✓
- ○ Error handling

✓ 2. DAG representation of a basic block allows 1/1

- [✓ Automatic detection of local common sub expressions] ✓
- ○ Automatic detection of induction variables
- ○ Automatic detection of loop variant
- ○ None of the above

✓ 3. Generation of intermediate code based on a abstract machine model is useful in compilers because — 1/1

- ○ it makes implementation of lexical analysis and syntax analysis easier
- ● syntax directed translation can be written for intermediate code generation. ✓
- ○ It enhances the portability of the front end of the compiler
- ○ it is not possible to generate code for real machines directly from high level language programs

✓ 4. An intermediate code form is — 1/1

- ○ Postfix notation
- ○ Syntax trees
- ○ Three address code
- ● All of these ✓

✓ 5. A compiler for a high level language that runs on one machine and produce code for different machine is called — 1/1

- ○ Optimizing compiler
- ○ One pass compiler
- ● Cross compiler ✓
- ○ Multipass compiler

✓ 6. Some code optimizations are carried out on the intermediate code because 1/1

- [✓ they enhance the portability of the compiler to other target processors] ✓
- program analysis is more accurate on intermediate code than on machine code
- the information from dataflow analysis cannot otherwise be used for optimization
- the information from the front end cannot otherwise be used for optimization

✓ 7. Which one of the following is FALSE? 1/1

- A basic block is a sequence of instructions where control enters the sequence at the beginning and exits at the end.
- Available expression analysis can be used for common subexpression elimination.
- Live variable analysis can be used for dead code elimination.
- [✓ x = 4 * 5 => x = 20 is an example of common subexpression elimination.] ✓

✓ 8. One of the purposes of using intermediate code in compilers is to 1/1

- make parsing and semantic analysis simpler.
- improve error recovery and error reporting.
- [✓ increase the chances of reusing the machine-independent code optimizer in other compilers.] ✓
- improve the register allocation.

✓ 9. Consider the following C code segment. for (i = 0, i<n; i++) { for (j=0; j<n; j++) { if (i%2) { x += (4*j + 5*i); y += (7 + 4*j); } }}Which one of the following is false? 1/1

- ○ The code contains loop invariant computation
- ○ There is scope of common sub-expression elimination in this code
- ○ There is scope of strength reduction in this code
- [✓ There is scope of dead code elimination in this code] ✓

✓ 10. What are the various types of three address statements 1/1

- ○ Assignment statement
- ○ copy statement
- ○ Assignment instruction
- [✓ All of the above] ✓