CS323 2025F Quiz #3

There are ten questions in this quiz.

1.Please put down your student ID. *

2.What is the primary purpose of a runtime environment in a compiler? *

○ To optimize the intermediate code.

○ To manage the execution of a program, including memory allocation and procedure linkages.

○ To perform lexical analysis and syntax analysis.

○ To generate the final target machine code.

3.Which memory allocation strategy is best suited for data whose size is unknown at compile time and whose lifetime may extend beyond the procedure that created it? *

○ Static Allocation

○ Stack Allocation

○ Heap Allocation

○ Register Allocation

4.What is the primary data structure used to manage memory for local variables and control information during procedure calls? *

○ Stack

○ Heap

○ Hash Table

○ Linked List

5.What does the term "activation tree" describe? *

○ The hierarchical structure of lexical scopes in the source program.

○ The tree of inheritance relationships among classes.

○ A tree representation of the possible dynamic call sequences of a program, where nodes are activations and edges represent calls.

○ The data structure used to implement a display for non-local variable access.

6.What is the primary goal of the code generation phase in a compiler? *

○ To produce an optimized intermediate representation.

○ To translate the intermediate representation into correct and efficient target machine code.

○ To perform lexical analysis and build the symbol table.

○ To resolve all memory addresses to absolute locations.

7.What is a "basic block" in control flow analysis? *

○ A sequence of instructions with no control flow in except to the first, and no control flow out except from the last.

○ The smallest unit of code that can be optimized independently.

○ A block of memory allocated for static data.

○ A machine instruction that alters the program counter.

8.Which of the following is a key advantage of using a machine-independent intermediate representation (IR) in a compiler? *

○ It allows the front-end and back-end to be developed independently, enabling retargeting.

○ It eliminates the need for a lexer and parser.

○ It can be directly executed by the target CPU for faster testing.

○ It is always more compact than either the source or target code.

9.Which of the following is a characteristic of a low–level Intermediate Representation (like LLVM IR) versus a high–level IR (like an Abstract Syntax Tree)? *

○ It is more closely tied to the source language syntax.

○ It exposes machine–specific details like an unlimited number of virtual registers and explicit control flow labels.

○ It retains high–level constructs like for–loops and class definitions.

○ It is not suitable for performing machine–independent optimizations.

10.Static Single Assignment (SSA) form is a powerful IR variant for optimizations. What is its defining characteristic? *

○ Every variable is assigned a static memory address at compile time.

○ Every variable is assigned exactly once in the code, and new versions (with subscripts) are created for multiple assignments.

○ It uses a stack–based evaluation model for all expressions.

○ It guarantees that every register is used only once.

11.What is the primary purpose of converting source code into a three–address code (TAC) IR? *

○ To create a representation that is close to assembly for easy code generation.

○ To break down complex expressions and statements into a sequence of simple, atomic instructions with at most one operator.

○ To perform sophisticated control flow analysis that is impossible on a parse tree.

○ To apply lexical analysis more efficiently.