1. What is the purpose of a type system in programming languages?

a) To enforce syntactic correctness
b) To specify program behavior
c) To ensure memory allocation
d) To facilitate error detection and recovery

Answer: b) To specify program behavior

Explanation: A type system specifies the allowed operations and behavior of variables and expressions in a programming language, ensuring consistency and correctness at compile time.

2. Which component of a type checker verifies whether expressions conform to the defined types?

a) Lexical analyzer
b) Parser
c) Semantic analyzer
d) Code generator

Answer: c) Semantic analyzer

Explanation: The semantic analyzer checks whether expressions and statements conform to the types defined by the language, ensuring type correctness.

3. What does equivalence of expressions in a type system refer to?

a) Similarity in syntax
b) Similarity in behavior
c) Similarity in memory allocation
d) Similarity in compilation time

Answer: b) Similarity in behavior

Explanation: Equivalence of expressions refers to expressions that have the same behavior or produce the same results, regardless of their specific syntactic forms.

4. Type conversion in programming languages involves:

a) Changing the syntax of a variable
b) Altering the behavior of an expression
c) Converting a variable from one type to another
d) Adjusting the memory allocation of a variable

Answer: c) Converting a variable from one type to another

Explanation: Type conversion involves changing the data type of a variable or expression to facilitate operations between different types.

5. What is overloading of functions and operations in programming languages?

a) Assigning multiple types to a single variable
b) Defining multiple functions with the same name but different parameter types
c) Limiting the number of operations that can be performed on a variable
d) Restricting the assignment of values to variables

Answer: b) Defining multiple functions with the same name but different parameter types

Explanation: Overloading allows multiple functions or operations to share the same name but differ in parameter types or arity.

6. Polymorphic functions in programming languages:

a) Are restricted to a single type
b) Can operate on values of different types
c) Can only be called once
d) Are limited to a specific number of parameters

Answer: b) Can operate on values of different types

Explanation: Polymorphic functions can accept arguments of different types, allowing for greater flexibility and code reuse.

7. What aspect of storage organization deals with how variables and data are allocated and managed during program execution?

a) Type checking
b) Storage allocation strategies
c) Error detection
d) Symbol table

Answer: b) Storage allocation strategies

Explanation: Storage allocation strategies determine how memory is allocated and managed for variables and data structures during program execution.

8. Which storage allocation strategy assigns memory to variables at compile time?

a) Dynamic allocation
b) Static allocation
c) Stack allocation
d) Heap allocation

Answer: b) Static allocation

Explanation: Static allocation assigns memory to variables at compile time, and their memory locations remain fixed throughout program execution.

9. Parameter passing refers to:

a) Allocating memory for function parameters
b) Passing arguments to a function
c) Passing return values from a function
d) Storing function parameters in a symbol table

Answer: b) Passing arguments to a function

Explanation: Parameter passing involves providing arguments to a function when calling it, which are then used as parameters within the function.

10. Dynamic storage allocation involves:

a) Allocating memory for variables at compile time
b) Allocating memory for variables at runtime
c) Allocating memory for functions
d) Allocating memory for constants

Answer: b) Allocating memory for variables at runtime

Explanation: Dynamic storage allocation involves allocating memory for variables and data structures during program execution, typically using functions like malloc() or new in languages like C and C++.

11. What purpose does a symbol table serve in a compiler or interpreter?

a) Storing function definitions
b) Managing memory allocation
c) Storing information about identifiers in the program
d) Handling runtime errors

Answer: c) Storing information about identifiers in the program

Explanation: A symbol table maintains information about identifiers (variables, functions, etc.) in the program, including their names, types, and scopes.

12. Error detection and recovery in compilers involves:

a) Detecting and correcting syntax errors
b) Detecting and correcting semantic errors
c) Detecting and recovering from runtime errors

d) Detecting and reporting errors during compilation

Answer: d) Detecting and reporting errors during compilation

Explanation: Error detection and recovery in compilers involve identifying errors in the source code during compilation and reporting them to the user for correction.

13. Ad-Hoc methods of error recovery in compilers refer to:

a) Using predefined algorithms to handle errors
b) Handling errors on a case-by-case basis
c) Ignoring errors during compilation
d) Automatically correcting errors

Answer: b) Handling errors on a case-by-case basis

Explanation: Ad-Hoc methods involve handling errors based on specific conditions or situations encountered during compilation, rather than relying on predefined algorithms.

14. Systematic methods of error recovery in compilers involve:

a) Ignoring errors and continuing compilation
b) Using predefined algorithms to handle errors
c) Correcting errors automatically
d) Reporting errors without attempting recovery

Answer: b) Using predefined algorithms to handle errors

Explanation: Systematic methods of error recovery involve following predefined algorithms or strategies to handle errors encountered during compilation.

15. Which of the following statements about type systems is true?

a) Type systems only enforce syntactic correctness
b) Type systems specify program behavior
c) Type systems don’t facilitate error detection
d) Type systems only ensure memory allocation

Answer: b) Type systems specify program behavior

Explanation: Type systems specify the behavior and constraints of variables and expressions in a program, ensuring consistency and correctness.

16. In parameter passing, which method involves passing the actual value of the argument to the function?

a) Pass by reference
b) Pass by value
c) Pass by pointer
d) Pass by name

Answer: b) Pass by value

Explanation: Pass by value involves passing a copy of the argument’s value to the function, leaving the original value unchanged.

17. Which of the following is NOT a storage allocation strategy?

a) Static allocation
b) Dynamic allocation
c) Stack allocation
d) Register allocation

Answer: d) Register allocation

Explanation: Register allocation is a technique used by compilers to optimize the use of processor registers, rather than a storage allocation strategy.

18. What does "equivalence of expressions" mean in the context of type systems?

a) Expressions that look the same
b) Expressions that have the same type
c) Expressions that produce the same result
d) Expressions that have the same memory address

Answer: c) Expressions that produce the same result

Explanation: Equivalence of expressions refers to expressions that produce the same result or behavior, regardless of their specific syntax or form.

19. Which of the following storage allocation strategies assigns memory to variables on a last-in, first-out basis?

a) Static allocation

b) Dynamic allocation
c) Stack allocation
d) Heap allocation

Answer: c) Stack allocation

Explanation: Stack allocation allocates memory for variables in a last-in, first-out manner, typically using a stack data structure.

20. What role does the semantic analyzer play in a compiler?

a) It checks for syntax errors
b) It optimizes the generated code
c) It verifies type correctness
d) It generates machine code

Answer: c) It verifies type correctness

Explanation: The semantic analyzer checks whether expressions and statements in the program conform to the types defined by the language, ensuring type correctness.

Related posts:

1. Introduction to Information Security
2. Introduction to Information Security MCQ
3. Introduction to Information Security MCQ
4. Symmetric Key Cryptography MCQ
5. Asymmetric Key Cryptography MCQ

6. Authentication & Integrity MCQ
7. E-mail, IP and Web Security MCQ