

Q1. Which keyword is used to define a package in GoLang?

- A. module
- B. package
- C. namespace
- D. import

Answer: B

Explanation: Every Go source file must start with the package keyword to define its spatial scope.

Q2. What is the default value of an uninitialized integer variable in a GoLang quiz?

- A. nil
- B. undefined
- C. 0
- D. -1

Answer: C

Explanation: In Go, variables are automatically assigned a "zero value" if not explicitly initialized; for integers, this is 0.

Q3. How do you declare a variable with an implicit type in GoLang?

- A. var x := 10
- B. x := 10
- C. int x = 10
- D. declare x = 10

Answer: B

Explanation: The short assignment operator := is used to declare and initialize a variable with type inference.

Q4. Which of the following is the correct way to import multiple packages in GoLang?

- A. import "fmt", "math"
- B. import ("fmt"; "math")
- C. import { "fmt" "math" }
- D. import ("fmt" "math")

Answer: D

Explanation: Multiple packages are imported using a factored import statement with parentheses.

Q5. In a GoLang MCQ test, which command is used to compile and run a program immediately?

- A. go build
- B. go execute
- C. go run
- D. go start

Answer: C

Explanation: The "go run" command compiles the source code and executes the resulting binary in one step.

Q6. What is the syntax for a single-line comment in GoLang?

- A. # comment
- B. // comment
- C. -- comment
- D. /* comment

Answer: B

Explanation: Go uses double forward slashes for single-line comments, similar to C++ and Java.

Q7. Which loop construct is natively supported in GoLang?

- A. while
 - B. do-while
-

- C. for
- D. foreach

Answer: C

Explanation: Go only has the "for" loop, which can be configured to function like a "while" loop.

Q8. What does the "defer" keyword do in a GoLang quiz context?

- A. Skips the next line of code
- B. Delays execution until the surrounding function returns
- C. Runs a function in a separate thread
- D. Stops the program immediately

Answer: B

Explanation: Defer ensures that a function call is performed later in a program's execution, usually for cleanup.

Q9. Which data type is used to represent a single Unicode character in GoLang?

- A. char
- B. byte
- C. rune
- D. unit

Answer: C

Explanation: A rune is an alias for int32 and is used to represent a Unicode code point.

Q10. How do you find the length of a string or slice in GoLang?

- A. len(x)
- B. x.length()
- C. size(x)
- D. count(x)

Answer: A

Explanation: The built-in len() function returns the number of elements in a slice or bytes in a string.

Q11. Which collection type in GoLang holds an unordered set of key-value pairs?

- A. Array
- B. Slice
- C. Map
- D. Struct

Answer: C

Explanation: Maps are Go's built-in associative data type, similar to dictionaries in other languages.

Q12. What is the entry point function for any executable GoLang program?

- A. start()
- B. init()
- C. main()
- D. execute()

Answer: C

Explanation: The main() function in the main package is the starting point of every Go executable.

Q13. How are goroutines started in a GoLang MCQ scenario?

- A. async function()
- B. thread function()
- C. go function()
- D. start function()

Answer: C

Explanation: The "go" keyword prefixed to a function call starts a lightweight thread called a goroutine.

Q14. Which symbol is used to access a pointer's value (dereferencing) in GoLang?

- A. &
- B. *
- C. @
- D. \$

Answer: B

Explanation: The asterisk symbol (*) is used to declare a pointer type and to dereference a pointer variable.

Q15. In GoLang, what is a "slice" compared to an "array"?

- A. Slices have a fixed size; arrays are dynamic
- B. Slices are dynamic; arrays have a fixed size
- C. Slices are for strings; arrays are for numbers
- D. They are exactly the same

Answer: B

Explanation: Arrays have a fixed length defined at compile time, while slices are flexible, windowed views into arrays.

Q16. Which operator is used for sending a value into a channel in GoLang?

- A. ->
- B. =>
- C. <-
- D. <<

Answer: C

Explanation: The channel operator <- is used to both send and receive data from channels.

Q17. What is the correct way to handle errors in GoLang?

- A. try-catch blocks
- B. throw-catch statements
- C. Returning an error as the last return value
- D. Using the "on-error" keyword

Answer: C

Explanation: Go handles errors by returning an "error" type value as the final result from a function.

Q18. Which tool is used to format GoLang source code automatically?

- A. go lint
- B. go tidy
- C. go fmt
- D. go clean

Answer: C

Explanation: "go fmt" is the standard tool that enforces a consistent coding style across all Go projects.

Q19. What keyword is used to create a custom data structure in GoLang?

- A. class
- B. object
- C. struct
- D. record

Answer: C

Explanation: The struct keyword is used to group different fields together into a single custom type.

Q20. Which of these is NOT a valid numeric type in GoLang?

- A. float32
-

- B. int64
- C. complex128
- D. decimal64

Answer: D

Explanation: Go supports specific bit-sized integers, floats, and complex numbers, but not a native decimal64 type.

Q21. What is the value of an uninitialized "bool" variable in a GoLang quiz?

- A. true
- B. false
- C. nil
- D. 0

Answer: B

Explanation: The zero value for a boolean variable in Go is false.

Q22. How do you declare a constant in GoLang?

- A. var const X = 1
- B. final X = 1
- C. const X = 1
- D. define X 1

Answer: C

Explanation: Constants are declared using the "const" keyword and must be assigned a value at declaration.

Q23. Which statement is used to exit a loop early in GoLang?

- A. exit
- B. stop
- C. break
- D. return

Answer: C

Explanation: The "break" statement terminates the execution of the innermost loop or switch statement.

Q24. What is the purpose of the "chan" keyword in GoLang?

- A. To create a change listener
- B. To define a channel for communication between goroutines
- C. To handle characters in a string
- D. To define a constant

Answer: B

Explanation: Channels are the pipes that connect concurrent goroutines, allowing them to synchronize and exchange data.

Q25. How do you define a function that returns two integers in GoLang?

- A. func test() (int, int) {}
- B. func test() [int, int] {}
- C. func test() int, int {}
- D. func test() { return int, int }

Answer: A

Explanation: Go supports multiple return values, which must be enclosed in parentheses in the function signature.

Q26. Which built-in function is used to create a slice, map, or channel?

- A. new()
- B. create()
- C. make()
- D. init()

Answer: C

Explanation: The `make()` function allocates and initializes internal data structures for slices, maps, and channels.

Q27. In a GoLang MCQ, what is the effect of using an underscore (`_`) in a variable assignment?

- A. It creates a private variable
- B. It acts as a blank identifier to discard a value
- C. It marks the variable as global
- D. It is a syntax error

Answer: B

Explanation: The blank identifier (`_`) is used when you need to ignore one of the return values of a function.

Q28. What is the difference between "new" and "make" in GoLang?

- A. new returns a pointer; make returns an initialized value
- B. make is for pointers; new is for slices
- C. they are identical in functionality
- D. new is only for arrays

Answer: A

Explanation: `new(T)` allocates zeroed storage and returns a `*T`; `make(T)` initializes slices, maps, and channels.

Q29. How do you check if a key exists in a map in GoLang?

- A. `if m.contains(key)`
- B. `if val, ok := m[key]; ok`
- C. `if m[key] != nil`
- D. `if exist(m[key])`

Answer: B

Explanation: Go's map access returns a second boolean value (often named 'ok') that indicates if the key exists.

Q30. What is a "method" in GoLang?

- A. A function that belongs to a class
- B. A function with a receiver argument
- C. A special type of goroutine
- D. A built-in system command

Answer: B

Explanation: A method is simply a function that has a "receiver" parameter between the 'func' keyword and the method name.

Q31. Which of these is used to group constants together with incrementing values?

- A. enum
- B. iota
- C. sequence
- D. auto

Answer: B

Explanation: `iota` is a special predeclared identifier used in `const` declarations to simplify definitions of incrementing numbers.

Q32. How do you append an element to a slice in GoLang?

- A. `slice.add(element)`
- B. `append(slice, element)`
- C. `slice += element`
- D. `push(slice, element)`

Answer: B

Explanation: The built-in `append()` function adds elements to the end of a slice and returns the updated slice.

Q33. What is the purpose of an interface in GoLang?

- A. To define a blueprint for structs using method signatures
- B. To create a GUI for the application
- C. To connect to a database
- D. To define global variables

Answer: A

Explanation: Interfaces define sets of method signatures; any type that implements those methods satisfies the interface.

Q34. Which keyword is used to handle multiple channel operations simultaneously?

- A. switch
- B. select
- C. multichan
- D. route

Answer: B

Explanation: The select statement lets a goroutine wait on multiple communication operations across different channels.

Q35. In a GoLang quiz, what happens if you try to modify a string?

- A. The string is updated
- B. A compile-time error occurs because strings are immutable
- C. A new string is created automatically
- D. The program crashes at runtime

Answer: B

Explanation: Strings in Go are immutable; you cannot change individual characters of a string after it is created.

Q36. What is the size of an "int" type in GoLang?

- A. Always 32-bit
- B. Always 64-bit
- C. Platform dependent (32 or 64-bit)
- D. Always 16-bit

Answer: C

Explanation: The "int" type is at least 32 bits wide, but it is typically 32 bits on 32-bit systems and 64 bits on 64-bit systems.

Q37. Which of the following is used to handle "panic" in GoLang?

- A. catch
- B. recover
- C. resume
- D. fix

Answer: B

Explanation: recover() is a built-in function that regains control of a panicking goroutine inside a deferred function.

Q38. How do you export a function from a package in GoLang?

- A. Use the "export" keyword
- B. Capitalize the first letter of the function name
- C. Use the "public" keyword
- D. Declare it in a separate .exp file

Answer: B

Explanation: In Go, visibility is determined by the case of the first letter: uppercase is exported (public), lowercase is unexported (private).

Q39. What is the capacity of a slice in GoLang?

- A. The number of elements it currently holds
 - B. The maximum size it can grow to without reallocation
 - C. The total size of the underlying array
 - D. The memory address of the first element
-

Answer: B

Explanation: Capacity (cap) is the number of elements in the underlying array, counting from the first element in the slice.

Q40. Which of these is a valid way to create an empty map in a GoLang MCQ?

- A. `m := map[string]int {}`
- B. `m := new(map[string]int)`
- C. `m := make(map[string]int)`
- D. Both A and C

Answer: D

Explanation: Maps can be initialized using either a map literal or the make function.

Q41. How does GoLang handle unused imports?

- A. It ignores them
- B. It gives a warning during compilation
- C. It results in a compilation error
- D. It automatically removes them from the source code

Answer: C

Explanation: Go is strict about code cleanliness; unused imports or unused local variables cause a build failure.

Q42. What is the result of 10 / 3 in GoLang if both are integers?

- A. 3.333
- B. 3
- C. 4
- D. Error

Answer: B

Explanation: Integer division in Go truncates the result toward zero.

Q43. Which command is used to initialize a new Go module?

- A. `go create mod`
- B. `go mod init`
- C. `go init module`
- D. `go start mod`

Answer: B

Explanation: "go mod init" creates a new go.mod file to track dependencies for your project.

Q44. What is a "nil" in GoLang?

- A. A zero value for pointers, interfaces, and slices
- B. A keyword for "nothing"
- C. An error type
- D. A boolean state

Answer: A

Explanation: nil is the zero value for pointers, interfaces, maps, slices, channels, and function types.

Q45. Which keyword is used to skip the current iteration of a loop in GoLang?

- A. skip
- B. pass
- C. continue
- D. next

Answer: C

Explanation: The continue statement begins the next iteration of the innermost for loop.

Q46. In GoLang, can a struct implement multiple interfaces?

- A. No, only one
- B. Yes, by implementing all required methods
- C. Only if the interfaces are in the same package
- D. Yes, using the "extends" keyword

Answer: B

Explanation: A type can satisfy multiple interfaces implicitly as long as it defines the methods required by each.

Q47. What is the purpose of the "time" package in GoLang?

- A. For CPU clock cycles
- B. To measure and display time and dates
- C. To limit goroutine execution
- D. For network latency

Answer: B

Explanation: The "time" package provides functionality for measuring and displaying time, including timers and tickers.

Q48. Which tool helps in detecting race conditions in GoLang programs?

- A. go test -race
- B. go detect
- C. go fix
- D. go check

Answer: A

Explanation: The Go toolchain includes a built-in race detector, activated by the -race flag during testing or building.

Q49. How do you convert an integer 'i' to a float64 in GoLang?

- A. float64(i)
- B. i.toFloat()
- C. (float64)i
- D. convert(i, float64)

Answer: A

Explanation: Go requires explicit type conversion using the T(v) syntax.

Q50. What is the "init" function used for in GoLang?

- A. To start the main program
- B. For package-level initialization before main() runs
- C. To define a constructor for a struct
- D. To reset variables

Answer: B

Explanation: init functions are called automatically when a package is initialized, even before the main function.

Q51. Which keyword is used to create a pointer to a variable in a GoLang quiz?

- A. *
- B. &
- C. ptr
- D. @

Answer: B

Explanation: The address-of operator (&) generates a pointer to its operand.

Q52. How do you declare a map with string keys and integer values in GoLang?

- A. map[int]string
 - B. map{string:int}
 - C. map[string]int
 - D. dictionary(string, int)
-

Answer: C

Explanation: The map declaration syntax is `map[KeyType]ValueType`.

Q53. What is the purpose of "range" in a GoLang for loop?

- A. To define the start and end of a loop
- B. To iterate over elements in a slice, map, or string
- C. To check if a value is within a limit
- D. To generate a sequence of numbers

Answer: B

Explanation: The range keyword provides an easy way to iterate over the items in a collection, yielding index and value.

Q54. Which function is used to print to the console with a newline in GoLang?

- A. `fmt.Print()`
- B. `fmt.Printf()`
- C. `fmt.Println()`
- D. `log.Write()`

Answer: C

Explanation: `fmt.Println()` formats its arguments and appends a newline character to the output.

Q55. In GoLang MCQ tests, what is "composition" in the context of structs?

- A. Inheriting from a base class
- B. Embedding one struct into another
- C. Merging two packages
- D. Converting a slice to an array

Answer: B

Explanation: Go uses struct embedding to achieve composition, allowing a struct to include fields/methods from another.

Q56. What is the zero value of an interface in GoLang?

- A. `{}`
- B. `nil`
- C. `empty`
- D. `undefined`

Answer: B

Explanation: An interface that has not been assigned a value is `nil`, meaning it has neither a type nor a value.

Q57. Which of these can be used as a key in a GoLang map?

- A. Slices
- B. Maps
- C. Functions
- D. Integers

Answer: D

Explanation: Only types that support the equality operator (`==`) can be used as map keys; slices and maps cannot.

Q58. How do you declare a function that takes a variable number of arguments (variadic) in GoLang?

- A. `func sum(nums []int)`
- B. `func sum(nums ...int)`
- C. `func sum(nums *int)`
- D. `func sum(args interface{})`

Answer: B

Explanation: Variadic functions use three dots (`...`) before the type to accept zero or more arguments.

Q59. What is the result of applying the "cap" function on an array in GoLang?

- A. 0
- B. Its fixed length
- C. The number of non-zero elements
- D. The total memory size

Answer: B

Explanation: For arrays, cap(a) is the same as len(a), returning the fixed size of the array.

Q60. Which keyword is used to send a value to a channel?

- A. send
- B. put
- C. <-
- D. push

Answer: C

Explanation: The arrow operator pointing toward the channel (ch <- value) is used to send data.

Q61. What is the standard way to name a Go source file?

- A. CamelCase.go
- B. snake_case.go
- C. ALLCAPS.go
- D. GoNames.go

Answer: B

Explanation: Go convention suggests using lowercase names with underscores if necessary (though short names are preferred).

Q62. Which of the following is used to perform a type assertion in GoLang?

- A. x.(T)
- B. (T)x
- C. x as T
- D. x.type(T)

Answer: A

Explanation: Type assertion x.(T) is used to extract the underlying value of an interface as a specific type T.

Q63. What is a "blank identifier" in GoLang?

- A. any
- B. _
- C. void
- D. null

Answer: B

Explanation: The underscore (_) is a special identifier used to discard values you don't need from assignments.

Q64. How do you create a pointer to a struct named "User" in GoLang?

- A. &User{}
- B. *User{}
- C. User.ptr()
- D. new(User{})

Answer: A

Explanation: Using the address-of operator & on a struct literal returns a pointer to that struct.

Q65. Which loop syntax is correct for an infinite loop in GoLang?

- A. for (true) {}
 - B. for ; ; {}
-

- C. for {}
- D. while (true) {}

Answer: C

Explanation: A "for" loop without any conditions or clauses is the standard way to write an infinite loop in Go.

Q66. What package provides the "Fatal" function to log and exit the program?

- A. fmt
- B. os
- C. log
- D. sys

Answer: C

Explanation: The log package's Fatal() function prints a message and calls os.Exit(1).

Q67. In a GoLang quiz, what is the default capacity of a slice created from make([]int, 5)?

- A. 0
- B. 5
- C. 10
- D. Undefined

Answer: B

Explanation: If only length is specified in make, the capacity defaults to the same value as the length.

Q68. Which keyword is used to prevent a struct from being copied?

- A. final
- B. static
- C. There is no such keyword; pointers are used instead
- D. private

Answer: C

Explanation: Go does not have a "non-copyable" keyword; developers use pointers or specific sync types to manage references.

Q69. How do you delete a key from a map in GoLang?

- A. m[key] = nil
- B. delete(m, key)
- C. m.remove(key)
- D. clear(m, key)

Answer: B

Explanation: The built-in delete() function removes the element associated with the given key from a map.

Q70. What does the "os.Args" slice contain?

- A. OS environment variables
- B. Command-line arguments including the program name
- C. List of running processes
- D. CPU architecture details

Answer: B

Explanation: os.Args is a slice of strings where the first element is the program name and others are input arguments.

Q71. Which operator is used for bitwise XOR in GoLang?

- A. ^
- B. &
- C. |
- D. !

Answer: A

Explanation: The caret symbol (^) represents the bitwise XOR operator in Go.

Q72. What is an "opaque" type in GoLang?

- A. A type that is visible but has no methods
- B. An unexported type used to hide implementation details
- C. A transparent interface
- D. A type with no name

Answer: B

Explanation: Unexported types (starting with lowercase) are opaque to other packages, hiding internal details.

Q73. How do you define a constant group in GoLang?

- A. const (...)
- B. define { ... }
- C. constants [...]
- D. grouped const { ... }

Answer: A

Explanation: Go allows grouping constant declarations together using parentheses to save typing "const" repeatedly.

Q74. In GoLang MCQ exams, what is the type of "error" in Go?

- A. A struct
- B. A built-in interface
- C. A string alias
- D. A pointer

Answer: B

Explanation: The error type is a built-in interface with a single method Error() string.

Q75. Which function is used to convert a string to an integer in GoLang?

- A. strconv.Atoi()
- B. fmt.ToInt()
- C. string.ParseInt()
- D. cast.Int()

Answer: A

Explanation: The "strconv" package's Atoi() (ASCII to Integer) is commonly used for string conversion.

Q76. Can GoLang functions return multiple values?

- A. No
- B. Only if they are pointers
- C. Yes, always
- D. Only inside a struct

Answer: C

Explanation: Go natively supports returning multiple values, frequently used for returning results and errors together.

Q77. What is the keyword used to check a condition in GoLang?

- A. check
- B. if
- C. verify
- D. when

Answer: B

Explanation: The if statement is the standard way to branch logic based on a boolean condition.

Q78. In GoLang, which operator is used for "not equal to"?

- A. <>
-

- B. !=
- C. !==
- D. NOT

Answer: B

Explanation: Similar to C-based languages, != is the inequality operator.

Q79. What happens if you call a method on a nil pointer receiver in GoLang?

- A. Always panics
- B. It is allowed as long as the method handles it
- C. Compile error
- D. Returns a zero value automatically

Answer: B

Explanation: Unlike many languages, Go allows calling methods on nil receivers; the method logic must check if the receiver is nil.

Q80. Which of these is a way to create a comment that spans multiple lines?

- A. // ... //
- B. /* ... */
- C. # ... #
- D. -- ... --

Answer: B

Explanation: Block comments in Go are enclosed within /* and */.

Q81. What does the "go build" command do?

- A. Runs the tests
- B. Compiles the package and creates an executable
- C. Installs the package in the bin folder
- D. Cleans the temporary files

Answer: B

Explanation: go build compiles the source code into an executable file but does not run it.

Q82. In a GoLang quiz, which statement describes "encapsulation" in Go?

- A. Using private and public keywords
- B. Using uppercase for exported and lowercase for unexported identifiers
- C. Using protected headers
- D. Using classes and objects

Answer: B

Explanation: Go uses capitalization to control the visibility (encapsulation) of identifiers outside the package.

Q83. Which built-in function returns the capacity of a channel?

- A. size()
- B. cap()
- C. len()
- D. limit()

Answer: B

Explanation: The cap() function returns the buffer capacity of a channel.

Q84. What is the keyword "range" usually used with?

- A. switch
- B. for
- C. select
- D. func

Answer: B

Explanation: range is used within a for loop to iterate over slices, maps, strings, and channels.

Q85. Which type is used for a 64-bit floating point number in GoLang?

- A. float
- B. double
- C. float64
- D. real64

Answer: C

Explanation: Go specifically uses float32 and float64; there is no "double" keyword.

Q86. How can you convert a slice of bytes back to a string?

- A. string(byteSlice)
- B. byteSlice.toString()
- C. cast(byteSlice, string)
- D. convert(byteSlice)

Answer: A

Explanation: Converting between a byte slice and a string is done using simple type conversion: string(b).

Q87. What is the purpose of the "sync" package in GoLang?

- A. For syncing files with the disk
- B. For basic synchronization primitives like Mutexes and WaitGroups
- C. To synchronize clock time
- D. To manage network connections

Answer: B

Explanation: The "sync" package provides tools to synchronize access to shared memory and manage goroutine execution.

Q88. Which operator is used for address-of in GoLang?

- A. *
- B. &
- C. @
- D. #

Answer: B

Explanation: The & operator returns the memory address of a variable.

Q89. In a GoLang MCQ context, what is a "buffered channel"?

- A. A channel that can hold a fixed number of values without a receiver
- B. A channel that clears its data every 10 seconds
- C. A channel used for high-speed streaming only
- D. A channel that only works with bytes

Answer: A

Explanation: Buffered channels have a capacity, allowing the sender to send multiple values before blocking.

Q90. Which of these is a correct map literal?

- A. map[string]int{"A": 1}
- B. map{string, int}("A", 1)
- C. map[string]int("A" = 1)
- D. {string:int}["A": 1]

Answer: A

Explanation: A map literal uses the type followed by curly braces containing key-value pairs.

Q91. What is the result of applying "len" to a map?

- A. The number of keys currently in the map
- B. The total capacity of the map
- C. The byte size of the map
- D. Always returns 0

Answer: A

Explanation: len(map) returns the number of key-value pairs currently stored in the map.

Q92. Which tool is used to download Go dependencies?

- A. go fetch
- B. go get
- C. go pull
- D. go install

Answer: B

Explanation: "go get" is the standard command for downloading and updating external Go packages.

Q93. What does "fallthrough" do in a Go switch statement?

- A. It jumps to the default case
- B. It executes the code block of the next case
- C. It exits the switch statement
- D. It restarts the switch

Answer: B

Explanation: Unlike C, Go switch cases break automatically; "fallthrough" is used to explicitly run the next case.

Q94. How do you define a constant that cannot be changed in GoLang?

- A. final X = 5
- B. var X = 5
- C. const X = 5
- D. static X = 5

Answer: C

Explanation: The const keyword defines a value that is fixed at compile time.

Q95. Which type is used for a string in GoLang?

- A. string
- B. String
- C. str
- D. []char

Answer: A

Explanation: The primitive type for sequences of characters in Go is "string" (lowercase).

Q96. What is the purpose of "iota" in a GoLang quiz?

- A. To define an interface
- B. To create a sequence of integer constants
- C. To handle errors
- D. To define a pointer

Answer: B

Explanation: iota is used in constant blocks to automatically increment the value of successive constants.

Q97. In GoLang, can you have a for loop without an initialization?

- A. No
 - B. Yes, by leaving it blank: for ; condition ; {}
 - C. Only in the main function
 - D. Only with slices
-

Answer: B

Explanation: Any of the three components of a Go for loop can be omitted.

Q98. Which of these is a valid function signature in GoLang?

- A. void test()
- B. func test(x int) int
- C. function test(x int)
- D. test := func()

Answer: B

Explanation: Go function signatures start with "func", then name, parameters, and finally the return type.

Q99. What does "import _" (blank import) do?

- A. It imports all subpackages
- B. It imports the package only for its side effects (like init functions)
- C. It results in a compile error
- D. It imports the package but renames it to blank

Answer: B

Explanation: A blank import executes the package's init() functions without making its identifiers available.

Q100. How do you create a WaitGroup in GoLang?

- A. sync.NewWaitGroup()
- B. var wg sync.WaitGroup
- C. make(sync.WaitGroup)
- D. wg := WaitGroup{}

Answer: B

Explanation: WaitGroups are typically declared as a variable of type sync.WaitGroup; they don't require explicit initialization with make or new.