

Set 05 Study Answers

Describe the purpose of each of the following keywords

1. public
Any method or variable can have this keyword in its declaration/definition. Anything declared as public is reachable from any method that has access to the class that contains the item declared as public.
2. private
Any method or variable can have this keyword in its declaration/definition. Anything declared as private is reachable only from the methods that are in the class that contains the item declared as public. If the item declared as public is an instance member, this item is available for any object (current object or otherwise).
3. static
Any method or variable can have this keyword in its declaration/definition. It means that there is one shared for all instances of the class. This means It should be accessed with the `ClassName.item` instead of through an individual instance of that class type. The alternative to static is instance. A method is indicated as being instance when the word static is omitted from the declaration/definition.
4. final
Used in the declaration of a variable (instance or static or local) to indicate that the value you are giving the variable during its declaration will remain its only value.
5. void
Used in the definition of a function (immediately before the function name) to indicate that the function will not return any value to the caller through the return value of the function. This means that it will be called as a statement rather than as an expression.

Define the following terms

6. stack
The runtime stack is used to track all local variables created. Called a stack because the variables put onto the stack are removed in the opposite order from how they were put onto the stack. Each method called has a “stack frame” which it

stacked on top of the stack frame for its caller. When the method is done, the stack frame for that method is removed from the stack. Only the 8 primitives and the reference type can be put onto the stack – everything else must be put onto the heap. When variables are added to the stack their value is random until they are assigned a value.

7. heap

The heap is where space is allocated for any object. Usually done with the key word “new” but the new can be implied (especially during initialization). The object as defined in the class comes into existence on the heap. All variables in that instance are initialized to their “0” value as it comes into existence.

8. reference

A reference is a pointer to a space on the heap. That reference can exist on the stack or in an object on the heap. The reference can be an alias or it can be the only pointer to that object or it can be null (meaning it is not currently pointing to an object).

9. variable declaration

When a variable space is initially created. For local variables this appears in the method but for instance and static variables it appears in the class definition. This gives place for the variable on the stack (for local variables) or on the heap within the object (for instance variables).

10. variable initialization

When the variable is given its initial value. For variables that come into existence on the heap there is a default initialization (if no explicit initialization is stated). For local variables, you will get an error if you try to use them before they are initialized.

11. heap space allocation

When space is requested using an explicit or implicit “new”, the object requested comes into existence on the heap.