MCQs BANK

Object Oriented Programming

Topic: Memory Management

MCQs BANK No.: 5

Instructions:

This MCQs Bank contains question and solution on adjacent(even-odd) pages. First try to solve the MCQ by yourself, then look for the solution.

Best viewed in “single page view” in PDF viewer.

MCQ No: 1

_____ is the process of automatically freeing objects that are no longer referenced by the program. Fill in the blank.

a) Polymorphism
b) Data Hiding
c) Garbage collection
d) Encapsulation

MCQ No: 1 (Solution)

Ans: c) Garbage collection

Explanation: The Java virtual machine's heap stores all objects created by a running Java application. Objects are created by the new keyword, but never freed explicitly by the code. Garbage collection is the process of automatically freeing objects that are no longer referenced by the program.

When an object is no longer referenced by the program, the heap space it occupies can be recycled so that the space is made available for subsequent new objects. The garbage collector must somehow determine which objects are no longer referenced by the program and make available the heap space occupied by such unreferenced objects.

MCQ No: 2

_____ occurs through the course of normal program execution, new objects are allocated, and unreferenced objects are freed such that free portions of heap memory are left in between portions occupied by live objects. Fill in the blank.

a) Garbage Collection
b) Heap fragmentation
c) Encapsulation
d) Polymorphism

MCQ No: 2 (Solution)

Ans: b) Heap fragmentation

Explanation: Heap fragmentation occurs through the course of normal program execution. New objects are allocated, and unreferenced objects are freed such that free portions of heap memory are left in between portions occupied by live objects. Requests to allocate new objects may have to be filled by extending the size of the heap even though there is enough total unused space in the existing heap. This will happen if there is not enough contiguous free heap space available into which the new object will fit.

MCQ No: 3

Which of the following are the advantage of garbage collection?

a) Garbage collection relieves you from the burden of freeing allocated memory.
b) It can make you more productive.
c) It helps to ensure program integrity.
d) All of the above.

MCQ No: 3 (Solution)

Ans: d) All of the above

Explanation: Garbage collection relieves you from the burden of freeing allocated memory. Knowing when to explicitly free allocated memory can be very tricky. Giving this job to the Java virtual machine has several advantages. First, it can make you more productive. When programming in non-garbage-collected languages you can spend many late hours (or days or weeks) chasing down an elusive memory problem. When programming in Java you can use that time more advantageously by getting ahead of schedule or simply going home to have a life. A second advantage of garbage collection is that it helps ensure program integrity. Garbage collection is an important part of Java's security strategy. Java programmers are unable to accidentally (or purposely) crash the Java virtual machine by incorrectly freeing memory.

MCQ No: 4

What are the disadvantages of a garbage-collection?

a) it adds an overhead that can affect program performance.
b) it requires more CPU time
c) programmers have less control on the memory management
d) all of the above

MCQ No: 4 (Solution)

Ans: d) all of the above

Explanation: A potential disadvantage of a garbage-collection is that it adds an overhead that can affect program performance. The Java virtual machine has to keep track of which objects are being referenced by the executing program, and finalize and free unreferenced objects on the fly. This activity will likely require more CPU time than would have been required if the program explicitly freed unnecessary memory. In addition, programmers in a garbage-collected environment have less control over the scheduling of CPU time devoted to freeing objects that are no longer needed.

MCQ No: 5

Which method is called by the garbage collector on an object when garbage collection determines that there are no more references to the object?

a) finalize()
b) free()
c) delete()
d) deallocate()

MCQ No: 5 (Solution)

Ans: a) finalize()

Explanation: finalize() is called by the garbage collector on an object when garbage collection determines that there are no more references to the object. A subclass overrides the finalize method to dispose of system resources or to perform other cleanup.

The Finalize method is used to perform cleanup operations on unmanaged resources held by the current object before the object is destroyed. The method is protected and therefore is accessible only through this class or through a derived class.

MCQ No: 6

Which of the following keyword can be used inside any method to refer to the current object.

a) this
b) new
c) static
d) final

MCQ No: 6 (Solution)

Ans: a) this

Explanation:

Sometimes a method will need to refer to the object that invoked it. To allow this, Java defines the this keyword. this can be used inside any method to refer to the current object.

That is, this is always a reference to the object on which the method was invoked. You can use this anywhere a reference to an object of the current class' type is permitted.

MCQ No: 7

If we call a method passing a value, it is known as call by value and the changes being done in the called method, is not affected in the calling method. True or False?

a) True
b) False

MCQ No: 7 (Solution)

Ans: a) True

Explanation:

There is only call by value in java, not call by reference(address). If we call a method passing a value, it is known as call by value. The changes being done in the called method, is not affected in the calling method.

In case of call by value original value is not changed.

MCQ No: 8

In case of _____ original value is changed if we made changes in the called method, that is if we pass object in place of any primitive value, original value will be changed. Fill in the blank.

a) call by value
b) call by reference
c) method overloading
d) method overriding

MCQ No: 8 (Solution)

Ans: b) call by reference

Explanation:

In case of call by reference original value is changed if we made changes in the called method.

If we pass object in place of any primitive value, original value will be changed.

MCQ No: 9

Which of the following modifier keyword makes that the programmer cannot change the value anymore?

a) final
b) static
c) abstract
d) volatile

MCQ No: 9 (Solution)

Ans: a) final

Explanation: The final modifier keyword makes that the programmer cannot change the value anymore. The actual meaning depends on whether it is applied to a class, a variable, or a method.

final Classes:- A final class cannot have subclasses. An example:

public final class MathConstants { ... }

This skeleton defines a class called MathConstants that is publicly accessible but cannot be subclassed.

Final Variables:-A final variable cannot be changed once it is initialized.

volatile:-The value of an attribute is not cached thread-locally, and is always read from the "main memory"

abstract: -Can only be used in an abstract class, can only be used on methods. The method does not have a body, for example *abstract void run();*. The body is provided by the subclass (inherited from).

MCQ No: 10

A _____ method cannot be overridden by subclasses. Fill in the blank.

a) public
b) default
c) final
d) volatile

MCQ No: 10 (Solution)

Ans: c) final

Explanation:

A final method cannot be overridden by subclasses. There are two reasons for introducing such a method:

1. Disallowing subclasses to change the meaning of the method;
2. Increasing efficiency by allowing the compiler to turn calls to the method into inline Java code.

MCQ No: 11

Can we OVERRIDE a static method?

a) Yes
b) No
c) Unpredictable
d) Machine dependent

MCQ No: 11 (Solution)

Ans: b) No

Explanation:

No, we cannot override static methods because method overriding is based on dynamic binding at runtime and the static methods are bonded using static binding at compile time. So, we cannot override static methods.

MCQ No: 12

Can we OVERLOAD a static method?

a) Yes
b) No
c) Unpredictable
d) Machine dependent

MCQ No: 12 (Solution)

Ans: a) Yes

Explanation:

The answer is Yes.

We can overload static methods. But remember that the method signature must be different.

MCQ No: 13

Can we overload the methods if they are only different by static keyword?

a) Yes
b) No
c) Unpredictable
d) Machine dependent

MCQ No: 13 (Solution)

Ans: b) No

Explanation:

The answer is No.

We cannot overload two methods if they differ only by static keyword. It give the error: method is already defined in the class.

MCQ No: 14

A variable or method that is shared by all instances of a class is called a class variable or class method. Such a variable or method in Java is declared by the _____ . Fill in the blank.

a) final Keyword
b) static keyword
c) volatile keyword
d) abstract keyword

MCQ No: 14 (Solution)

Ans: b) static keyword

Explanation:

A variable or method that is shared by all instances of a class is called a class variable or class method. You recognize such a variable in Java by the static keyword in the declaration. A class variable will instantiate only one copy of the variable for the whole class instead of a separate copy for each instance of a class. A class variable belongs to a class, not to an instance of the class.

MCQ No: 15

The Java programming language allows you to define a class within another class. Such a class is called a _____. Fill in the blank.

a) inside class
b) nested class
c) core class
d) static class

MCQ No: 15 (Solution)

Ans: b) nested class

Explanation: The Java programming language allows you to define a class within another class. Such a class is called a nested class and is illustrated here:

```
class OuterClass {
 ...
   class NestedClass {
   ... }
 }
```

A nested class is a member of its enclosing class. Non-static nested classes (inner classes) have access to other members of the enclosing class, even if they are declared private. Static nested classes do not have access to other members of the enclosing class. As a member of the OuterClass, a nested class can be declared private, public, protected, or package private.

MCQ No: 16

Which of the following are the reasons of using nested classes?

a) It is a way of logically grouping classes that are only used in one place.
b) It increases encapsulation.
c) Nested classes can lead to more readable and maintainable code.
d) All of these.

MCQ No: 16 (Solution)

Ans: d) All of these.

Explanation:

Logical grouping of classes— If a class is useful to only one other class, then it is logical to embed it in that class and keep the two together. Nesting such "helper classes" makes their package more streamlined.

Increased encapsulation— Consider two top-level classes, A and B, where B needs access to members of A that would otherwise be declared private. By hiding class B within class A, A's members can be declared private and B can access them. In addition, B itself can be hidden from the outside world.

More readable, maintainable code— Nesting small classes within top-level classes places the code closer to where it is used.

MCQ No: 17

If you declare an inner class(with a class name) within the body of a method. Such a class is known as a _____. Fill in the blank.

a) anonymous inner class.
b) local inner class
c) global inner class.
d) static inner class

MCQ No: 17 (Solution)

Ans: b) local inner class

Explanation:

There are two additional types of inner classes. You can declare an inner class within the body of a method. Such a class is known as a local inner class. You can also declare an inner class within the body of a method without naming it. These classes are known as anonymous inner classes.

MCQ No: 18

You can also declare an inner class within the body of a method without naming it. Such class is known as _____. Fill in the blank.

a) anonymous inner class.
b) local inner class.
c) global inner class.
d) static inner class

MCQ No: 18 (Solution)

Ans: a) anonymous inner class.

Explanation:

There are two additional types of inner classes. You can declare an inner class within the body of a method. Such a class is known as a local inner class. You can also declare an inner class within the body of a method without naming it. These classes are known as anonymous inner classes.

MCQ No: 19

You can use the access specifiers — private, public, and protected — to restrict access to inner classes. True or False

a) True
b) False

MCQ No: 19 (Solution)

Ans: a) True

Explanation: You can use the same modifiers for inner classes that you use for other members of the outer class. For example, you can use the access specifiers — private, public, and protected — to restrict access to inner classes, just as you do to other class members.

MCQ No: 20

Which of the following statements are correct?

a) To instantiate an outer class, you must first instantiate the inner class.
b) To instantiate an inner class, you must first instantiate the outer class.
c) To instantiate an local inner class, you must first instantiate the local outer class.
d) To instantiate an anonymous class, you must first instantiate the local outer class.

MCQ No: 20 (Solution)

Ans: b) To instantiate an inner class, you must first instantiate the outer class.

Explanation:

To instantiate an inner class, you must first instantiate the outer class. Then, create the inner object within the outer object with this syntax:

OuterClass.InnerClass innerObject = outerObject.new InnerClass(); .