# Operators – Answers

Question

d Prints: 2, 2, -3, 3, 4,

The expression can be simplified as follows: j = 2 + 2 + -3 + 3 = 4.

The original expression is as follows: j = ++i + i++ + -i + i++.

Simplification step one is to evaluate the unary expressions from left to right: j = 2 + 2 + -3 + 3.

Simplification step two is to complete the evaluation of the simplified expression: j = 4.

Question

d Prints: false,true,true

The right hand operand of the conditional or operator is evaluated only if the left hand operand is false. The right hand operand of the conditional and operator is only evaluated if the left hand operand is true. In this case, the left hand operand of the conditional or operator is false, so the right hand operand must also be evaluated. The left hand operand of the conditional and operator is the result of the conditional or expression, true, so the right hand operand is evaluated.

Question

a d e

Encapsulation is a form of data hiding. Encapsulation helps to protect data from corruption. Encapsulation allows for changes to the internal design of a class while the public interface remains unchanged. A tightly encapsulated class does not allow direct public access to the internal data model. Instead, access is permitted only through accessor (i.e. get) and mutator (i.e. set) methods. The additional time required to work through the accessor and mutator methods typically slows execution speed. Encapsulation is a form of data hiding. A tightly encapsulated class does not allow public access to any data member that can be changed in any way; so encapsulation helps to protect internal data from the possibility of corruption from external influences. The mutator methods can impose contraints on the argument values. If an argument falls outside of the acceptable range, then a mutator method could throw an IllegalArgumentException. The internal design of a tightly encapsulated class can change while the public interface remains unchanged. An immutable class is always tightly encapsulated, but not every tightly encapsulation class is immutable.

Question

e continue, finalize, break, package

The word finalize is the name of a method of the Object class: It is not a keyword. The words continue, goto, package and synchronized are all Java keywords.

Question

**b The null value**

All arrays of reference type variables such as String are initialized to the special null value.

Question

c

The default accessibility for members is more restrictive than protected accessibility, but less restrictive than private. Members with default accessibility are only accessible within the class itself and from classes in the same package. Members with protected accessibility are in addition accessible from subclasses anywhere. Members with private accessibility are only accessible within the class itself.

Question

f Compile-time error

Variables declared inside of a block or method are local variables; they are not automatically initialized. The compiler will generate an error as a result of the attempt to access the local variables before a value has been assigned.

Question

c Prints: 0null

The numeric sum of variables a, b, c, d and e is zero. The zero is converted to a String and concatenated with s.

Question

b Prints: 2,3

Variables of primitive type are passed to methods by value: Only a copy of the value of the variable is passed to the method. While the method works with a local copy of the variable, the original variable remains unchanged by any actions performed on the method parameter. For that reason, method m1 does not change the value of the variable y in the main method. However, method m1 does have direct access to the class variable x and the content of the class variable is modified by method m1.

Question

d None of the above

The compound assignment operators include an implicit cast to the type of the left hand operand. The expression at line 3, b += l, does not require an explicit cast to convert the right hand operand from type long to type byte.

Question

j Compile-time error.

The precedence of the cast operator is higher than the precedence of the addition operator, so the cast applies only to variable a (and not to the result of the addition). Binary numeric promotion causes the byte variables a and b to be promoted to type int before the addition operation, and the result of the addition is also of type int. The attempt to assign the int result to the byte variable e generates a possible loss of precision error.

Question

**a, c**

Abstract classes can contain both final methods and non-abstract methods. Non-abstract classes cannot, however, contain abstract methods. Nor can abstract classes be final. Only methods can be declared native.

Question 13

b Prints: 1,3

The method m1 is invoked by the method invocation expression m1(i1, i2). The argument i1 denotes a local variable of type int[] that is declared in the main method. The value of the argument is a reference to the array, and the argument value is used to initialize the method parameter i1 of method m1. Inside the body of m1, the expression i1 = i2 sets the value of parameter i1 to the value of parameter i2, but the change in the value of the parameter i1 does not change the original argument value or the local variable i1 of the main method that the argument denotes. Similarly, the assignment expression i2 = i3 in method m1 does not change the value of the local variable i1 declared in the main method.

Question

c Prints: BFCTAT

The right operand of the conditional or operator is evaluated only if the left hand operand is false. In this case, the left operand of the first conditional or operator is false so the right hand operand is evaluated. No further evaluation of the expression is necessary so the right hand operand of the second conditional or operator is not evaluated.

Question

f Prints: true,false,true

The positive infinity of type float is promoted to the positive infinity of type double. NaN is not equal to anything including itself.

Question

c Prints: 195ab

Both operands of the first addition operator are promoted from type char to int, and are evaluated as integral numeric values. The right hand operand of the second addition operator is of type String, so the result of the first addition operator is converted to type String, and is concatenated with the right hand operand. As evaluation of the expression continues from left to right, the remaining operands are also converted to type String.

Question 17

a Prints: true

The literal, 'a', is promoted to type int; and is then multiplied by the value of the left operand, 4. If one of the two operands of a numeric expression is of type int, then the other operand will be promoted to type int if it is of type short, char or byte.

Question

c Prints: 3,1

Method m1 is not able to change the value of the local variables that are declared in the main method and serve as the arguments in the method invocation expression. However, method m1 is able to modify the contents of the arrays that are referenced by the method parameters.

Question

c

A class is uninstantiable if the class is declared abstract. The declaration of an abstract method cannot provide an implementation. The declaration of a non-abstract method must provide an implementation. If any method in a class is declared abstract, then the class must be declared abstract. Definition (d) is not valid since it omits the class keyword.

Question

d Prints: 1,2,3,4,5,3

The expression can be simplified as follows: j = 1 + ((2 \* 3) % 4) + 5 = 8.

The original expression is as follows: j = ++i + ++i \* ++i % ++i + ++i.

Step one: Evaluate the unary expressions from left to right: j = 1 + 2 \* 3 % 4 + 5.

Step two: Add parentheses to indicate operator precedence: j = 1 + ((2 \* 3) % 4) + 5.

Step three. Evaluate the inner most parentheses: j = 1 + (6 % 4) + 5.

Repeat step three: j = 1 + 2 + 5.

Repeat step three: j = 8.

The argument of the print expression is: j%5, and the result is 8 % 5, i.e. 3.

Question

f Prints: true,false,true

The left operand of the instanceof operator must be null or a reference to an instance of an Object or a subclass of Object. The right operand of the instanceof operator must be a class type, interface type or array type. If the left operand is a reference to an instance of the type specified by the right operand or if the left operand is a reference to an instance of a subclass of the type specified by the right operand, then instanceof returns true.

Question

b Prints: FFT

The expression used to assign variable b1 is equivalent to the expression used to assign variable b2. The results demonstrate that the conditional operator (?:) groups from right-to-left.