Java Questions & Answers – Bitwise Operators

This section of our 1000+ Java MCQs focuses on Bitwise operators of Java Programming Language.

1. Which of these is not a bitwise operator?  
   a) &  
   b) &=  
   c) |=  
   d) <= [expand title=""]

**Answer:d**

Explanation: <= is a relational operator. [/expand]

2. Which operator is used to invert all the digits in binary representation of a number? a) ~ b) <<< c) >>>  
d) ^  
**Answer:a**Explanation: Unary not operator, ~, inverts all of the bits of its operand in binary representation.

3. On applying Left shift operator, <<, on an integer bits are lost one they are shifted past which position bit? a) 1 b) 32 c) 33 d) 31 [expand title=""] Answer: d Explanation: The left shift operator shifts all of the bite in a value to the left specified number of times. For each shift left, the high order bit is shifted out and lost, zero is brought in from right. When a left shift is applied to an integer operand, bits are lost once they are shifted past the bit position 31. [/expand] 4. Which right shift operator preserves the sign of the value? a) << b) >>  
c) <<= d) >>=  
**Answer: b**Explanation: None.

5. Which of these statements are incorrect?  
a) The left shift operator, <<, shifts all of the bite in a value to the left specified number of times. b) The right shift operator, >>, shifts all of the bite in a value to the right specified number of times.  
c) The left shift operator can be used as an alternative to multiplying by 2.  
d) The right shift operator automatically fills the higher order bits with 0.  
**Answer: d**Explanation: The right shift operator automatically fills the higher order bit with its previous contents each time a shift occurs. This also preserves the sign of the value.

6. What is the output of this program?

1. **class** bitwise\_operator {
2. **public** **static** **void** main(String args[])
3. {
4. **int** var1 = 42;
5. **int** var2 = ~var1;
6. System.out.print(var1 + " " + var2);
7. }
8. }

a) 42 42  
b) 43 43  
c) 42 -43  
d) 42 43  
**Answer:c**Explanation: Unary not operator, ~, inverts all of the bits of its operand. 42 in binary is 00101010 in using ~ operator on var1 and assigning it to var2 we get inverted value of 42 i:e 11010101 which is -43 in decimal.  
output:  
$ javac bitwise\_operator.java  
$ java bitwise\_operator  
42 -43

7. What is the output of this program?

1. **class** bitwise\_operator {
2. **public** **static** **void** main(String args[])
3. {
4. **int** a = 3;
5. **int** b = 6;
6. **int** c = a | b;
7. **int** d = a & b;
8. System.out.println(c + " " + d);
9. }
10. }

a) 7 2  
b) 7 7  
c) 7 5  
d) 5 2  
**Answer: a**Explanation: And operator produces 1 bit if both operand are 1. Or operator produces 1 bit if any bit of the two operands in 1.  
output:  
$ javac bitwise\_operator.java  
$ java bitwise\_operator  
7 2

8. What is the output of this program?

1. **class** leftshift\_operator {
2. **public** **static** **void** main(String args[])
3. {
4. **byte** x = 64;
5. **int** i;
6. **byte** y;
7. i = x << 2;
8. y = (**byte**) (x << 2)
9. System.out.print(i + " " + y);
10. }
11. }

a) 0 64  
b) 64 0  
c) 0 256  
d) 256 0  
**Answer:d**Explanation: None.  
output:  
$ javac leftshift\_operator.java  
$ java leftshift\_operator  
256 0

9. What is the output of this program?

1. **class** rightshift\_operator {
2. **public** **static** **void** main(String args[])
3. {
4. **int** x;
5. x = 10;
6. x = x >> 1;
7. System.out.println(x);
8. }
9. }

a) 10  
b) 5  
c) 2  
d) 20  
**Answer: b**Explanation: Right shift operator, >>, devides the value by 2.  
output:  
$ javac rightshift\_operator.java  
$ java rightshift\_operator  
5

10. What is the output of this program?

1. **class** Output {
2. **public** **static** **void** main(String args[])
3. {
4. **int** a = 1;
5. **int** b = 2;
6. **int** c = 3;
7. a |= 4;
8. b >>= 1;
9. c <<= 1;
10. a ^= c;
11. System.out.println(a + " " + b + " " + c);
12. }
13. }

a) 3 1 6  
b) 2 2 3  
c) 2 3 4  
d) 3 3 6  
**Answer: a**Explanation: None.  
output:  
$ javac Output.java  
$ java Output  
3 1 6