Java Questions & Answers – Integer and Floating Data Types

This Section of our 1000+ Java MCQs focuses on Integer and Floating Datatypes of Java Programming Language.

1. What is the range of data type short in Java?  
a) -128 to 127  
b) -32768 to 32767  
c) -2147483648 to 2147483647  
d) None of the mentioned  
 **Answer:b**Explanation:Short occupies 16 bits in memory. Its range is from -32768 to 32767.

2. What is the range of data type byte in Java?  
a) -128 to 127  
b) -32768 to 32767  
c) -2147483648 to 2147483647  
d) None of the mentioned

**Answer:a**Explanation:Byte occupies 8 bits in memory. Its range is from -128 to 127.

3. Which of the following are legal lines of Java code?  
1. int w = (int)888.8;  
2. byte x = (byte)100L;  
3. long y = (byte)100;  
4. byte z = (byte)100L;  
a) 1 and 2  
b) 2 and 3  
c) 3 and 4  
d) All statements are correct.

**Answer: d**Explanation:Statements (1), (2), (3), and (4) are correct. (1) is correct because when a floating-point number (a double in this case) is cast to an int, it simply loses the digits after the decimal.(2) and (4) are correct because a long can be cast into a byte. If the long is over 127, it loses its most significant (leftmost) bits.(3) actually works, even though a cast is not necessary, because a long can store a byte.

4. An expression involving byte, int, and literal numbers is promoted to which of these?  
a) int  
b) long  
c) byte  
d) float

**Answer: a**Explanation:An expression involving bytes, ints, shorts, literal numbers, the entire expression is promoted to int before any calculation is done.

5. Which of these literals can be contained in a data type float variable?  
a) 1.7e-308  
b) 3.4e-038  
c) 1.7e+308  
d) 3.4e-050

**Answer: b**Explanation:Range of data type float is 3.4e-038 to 3.4e+308.

6. Which data type value is returned by all transcendental math functions?  
a) int  
b) float  
c) double  
d) long

**Answer:c**Explanation:None.

7. What is the output of this program?

1. **class** average {
2. **public** **static** **void** main(String args[])
3. {
4. **double** num[] = {5.5, 10.1, 11, 12.8, 56.9, 2.5};
5. **double** result;
6. result = 0;
7. **for** (**int** i = 0; i < 6; ++i)
8. result = result + num[i];
9. System.out.print(result/6);
11. }
12. }

advertisements

a) 16.34  
b) 16.566666644  
c) 16.46666666666667  
d) 16.46666666666666

**Answer:c**Explanation:None.  
output:  
$ javac average.java  
$ java average  
16.46666666666667

8. What is the output of this program?

1. **class** conversion {
2. **public** **static** **void** main(String args[])
3. {
4. **double** a = 295.04;
5. **int** b = 300;
6. **byte** c = (**byte**) a;
7. **byte** d = (**byte**) b;
8. System.out.println(c + " " + d);
9. }
10. }

a) 38 43  
b) 39 44  
c) 295 300  
d) 295.04 300

**Answer:b**Explanation:Type casting a larger variable into a smaller variable results in modulo of larger variable by range of smaller variable. b contains 300 which is larger than byte’s range i:e -128 to 127 hence d contains 300 modulo 256 i:e 44.  
output:  
$ javac conversion.java  
$ java conversion  
39 44

9. What is the output of this program?

1. **class** increment {
2. **public** **static** **void** main(String args[])
3. {
4. **int** g = 3;
5. System.out.print(++g \* 8);
6. }
7. }

advertisements

a) 25  
b) 24  
c) 32  
d) 33

**Answer:c**Explanation:Operator ++ has more preference than \*, thus g becomes 4 and when multiplied by 8 gives 32.  
output:  
$ javac increment.java  
$ java increment  
32

10. What is the output of this program?

1. **class** area {
2. **public** **static** **void** main(String args[])
3. {
4. **double** r, pi, a;
5. r = 9.8;
6. pi = 3.14;
7. a = pi \* r \* r;
8. System.out.println(a);
9. }
10. }

a) 301.5656  
b) 301  
c) 301.56  
d) 301.56560000

**Answer:a**Explanation:None.  
output:  
$ javac area.java  
$ java area  
301.5656