**Integer And Floating Data Type:**

**1. What is the range of short data type 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 byte data type 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) int**

**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 float data type variable?**

a) -1.7e+308

b) -3.4e+038

c) +1.7e+308

d) -3.4e+050

**Answer: b**Explanation: Range of float data type is -(3.4e38) To +(3.4e38)

**6. Which data type value is returned by all transcendental math functions?**

a) int

b) float

c) double

d) long

**Answer) c)**

**Transcendental Math Functions**

Java provides two transcendental math functions. The first raises Euler's constant to the power of the value of an argument. The second returns the natural logarithm of an argument.

public static double exp(double a)

public static double log(double a)

exp() returns the value e a , Euler's number ( e ) raised to the power of the input argument. If the input argument is NaN , the return value is NaN . If the input argument is negative infinity, zero is returned. If the input argument is positive infinity, positive infinity is returned.

log() returns the natural logarithm of the input argument. If the argument is less than zero or NaN , the return value is NaN .

**7. What is the output of this program?**

class average {

public static void main(String args[])

{

double num[] = {5.5, 10.1, 11, 12.8, 56.9, 2.5};

double result;

result = 0;

for (int i = 0; i < 6; ++i)

result = result + num[i];

System.out.print(result/6);

}

}

a) 16.34

b) 16.566666644

c) 16.46666666666667

d) 16.46666666666666

**Answer: c**

**8. What will be the output of these statement?**

class output {

public static void main(String args[])

{

double a, b,c;

a = 3.0/0;

b = 0/4.0;

c=0/0.0;

System.out.println(a);

System.out.println(b);

System.out.println(c);

}

}

a) Infinity

b) 0.0

c) NaN

d) all of the mentioned

**Answer: d) All of the mentioned**

Explanation: For floating point literals, we have constant value to represent (10/0.0) infinity either positive or negative and also have NaN (not a number for undefined like 0/0.0),

Now, first one will generate Infinity

Second one will generate 0.0

Third one will generate NaN

Now, In this case, the whole expression is changed to floating point before evaluation. So, it wont throw arithmatic point exception.

But, if for some case, the expression is int before evaluation, the similar cases would generate ArithmaticException.

**Consider the following example:**

class OutputExample

{

public static void main(String args[])

{

double a, b,c;

a = 3/0;

System.out.println(a);

}

}

**This will generate something like following:**

Exception in thread "main" java.lang.ArithmeticException: / by zero

at OutputExample.main(OutputExample.java:6)

**9. What is the output of this program?**

class increment {

public static void main(String args[])

{

int g = 3;

System.out.print(++g \* 8);

}

}

a) 25

b) 24

c) 32

d) 33

**Answer) c) 32**

Explanation: Operator ++ has more preference than \*, thus g becomes 4 and when multiplied by 8 gives 32.

**10. What is the output of this program?**

class area {

public static void main(String args[])

{

double r, pi, a;

r = 9.8;

pi = 3.14;

a = pi \* r \* r;

System.out.println(a);

}

}

a) 301.5656

b) 301

c) 301.56

d) 301.56560000

**Answer: a**