Relational operators

If **d1** and **d2** are doubles with valid values, what is wrong with the following expression?

```
"answer = " + (d1 < d2)
```

return boolean values true/false

Select one:

- a. Can't add a String and a boolean
- b. Can't concatenate a String and a boolean
- c. Can't compare a String and a boolean
- d. Can't compare a String and a double
- e. If you think nothing is wrong with the expression, select this option

If the value of x is 20, what does this expression evaluate in Java?

```
0<= x < 30
```

Boolean int

Select one:

- a. false
- b. true
- c. It will not compile

What does the following expression evaluate to?

```
boolean flag="abcd".compareTo("abCd ") > 0;
```

Select one:

- a. The expression has a syntax error
- b. true
- c. false

.compareTo() & .compareToIgnoreCase(): return int

Unicode order:
space<1<2<3..<9<…<A<B<C<…Z<a<b<c<…<z

Operators

Which of the following is NOT a Java operator?

Remember, for this question and other questions in this quiz: "When in doubt, try it out (in IntelliJ)".

Select one or more:

- [ ] a. -- decrement operator
- [x] b. <-
- [x] c. ===
- [ ] d. != relational operator is not equal to
- [ ] e. || logical OR
- [ ] f. <= arithmetic operator (numeric comparison)
- [ ] g. ! boolean operator NOT
- [ ] h. -= subtraction assignment operator

Arithmetic operator & equality operator

If the following expression was evaluated for each integer between **10** and **150** inclusive how many times would it return true?

first one: 17
last one: 17*8=136

```
i % 17 == 0
```

Select one:

- ( ) a. 10
- ( ) b. 9
- (•) c. 8
- ( ) d. 1
- ( ) e. 0
- ( ) f. 7

Arithmetic operator

What does the following expression evaluate to?

```
(double) (24 / ((double) 5))
```

24/5->4
24/(double)5->4.8

Select one:

- a. 4
- b. 5
- c. 4.0
- d. 3.5
- e. 5.0
- f. 4.8

Variable scope

If a statement block is nested inside another statement block which of the following is true?

Select one:

- a. The variables in the inner block are in scope in the outer block but the reverse is not true
- b. The variables of the outer block are in scope in the inner block but the reverse is not true
- c. The variables in the outer block are in scope in the inner block and the reverse is also true
- d. If you think nesting blocks like this produces a syntax error in Java, select this option

Pre/post increment

What is the output of the following piece of code?

```
int x = 4, y = 3;
boolean b = ++x < y++ || ++x < y++;
System.out.println(x+","+y+","+b);
```

Select one:

a. 5,6,false

b. 6,5,false

c. 5,4,false

d. 4,3,false

e. 4,3,true

f. 5,4,true

What left operand **value** and right operand **value** respectively does the multiplication operator operate on in the following expression? You can assume i has the initial value of 5.

i-- * i--

Hint: try it out in IntelliJ and work backwards to deduce what must have happened.

Select one:

a. 5, 5

b. 4, 3

c. 3, 3

d. 5, 4

e. 4, 4

Math class

Which of the following is NOT true about the **Math** class?

Select one or more:

- [x] a. A Math object can be instantiated
- [ ] b. It is not a recipe for a thing/concept instance
- [ ] c. Its methods perform mathematical functions
- [ ] d. It contains public mathematical constants Math.E Math.PI
- [ ] e. It never needs importing in a package in Java class library called java.lang
  double a=Math.sqrt(a); double power=Math.pow(b,2);
- [x] f. It does not provide operations such as roots and powers as these basic mathematical operations (along with +, - *, /) are provided by the Java language