1. Choose the appropriate data type for this value: **1 / 1 point**

5.5

- ○ int
- ◉ double
- ○ String
- ○ None

✓ **Correct**
The appropriate data type for this value 5.5 is double

2. What will the following code result in? **1 / 1 point**

int num = 6.7;

- ○ num being 6
- ◉ compile error
- ○ nothing
- ○ num being 6.7

3. Which of the following types is not a primitive data type in Java? **1 / 1 point**

- ◉ String
- ○ int
- ○ boolean
- ○ char

✓ **Correct**
String is not a primitive data type in Java.

4. Which statement is correct in syntax? **1 / 1 point**

- ◉ int myInt = 2;
- ○ myInt = 2;
- ○ int myInt = 3
- ○ int myInt = "4";

5. What will the code below output? **1 / 1 point**

```
1  System.out.println(5 > 4);
2
```

- ○ True
- ◉ true
- ○ False
- ○ false

✓ **Correct**
It will output true because 5 > 4 is "true" and the String "true" in Java is *true*

6. If you want your condition to depend on both conditions being true, what's the proper notation? **1 / 1 point**

- ◉ &&
- ○ ||
- ○ and
- ○ !

✓ **Correct**
&& is the proper notation if you want your condition to depend on both conditions being true

7. What's the correct syntax for Java's main method? **1 / 1 point**

- ○ public void main()
- ◉ public static void main(String[] args)
- ○ public static void main(string[] args)
- ○ public static void main()

8. If you want to define a char first, and then a String, what's the correct type of quotation marks (and order) to use? **1 / 1 point**

- ○ First "", then ""
- ◉ First '', then ""
- ○ First '', then '
- ○ First "", then '

✓ **Correct**

'' is used to define a char while "" is used to define a String

9. What will the code below output? **1 / 1 point**

```
1  String a = "Hello";
2  char b = '!';
3  int c = 0;
4  System.out.println(a + b + c);
5
```

- ○ Compile error
- ◉ "Hello!0"
- ○ "Hello"
- ○ "0"

✓ **Correct**

It will output "Hello!0" because the plus sign (+) is a concatenation operator in Java.

10. What is used in Java to surround code blocks?

1 / 1 point

- ○ :
- ○ Indentation
- ◉ {}
- ○ ()

✓ **Correct**

{} is used in Java to surround code blocks