

Read each question carefully. When writing code, note that not all questions require a complete class, or even a complete method.

When a method is called, who is responsible for ensuring that the postcondition is valid?

- A. The person who is using the program.
- B. The programmer who called the method.
- C. The programmer who wrote the method.
- D. The programmer who implemented the Java Runtime System.

Why does a run-time analysis usually count arithmetic and other operations instead of the number seconds required for the program to run?

What are boundary values (with regards to testing programs)?

Consider this code that creates some Location objects with coordinates x=10 and y=20:

```
Location hq = new Location(10,20);
NamedLocation secretBase = new NamedLocation(10,20,"blue team base");
...
```

After this code executes, what are the values of these boolean expressions?

a==b	a.equals(b)
b==a	b.equals(a)

Consider the Java code below & determine what the final output is:

```
int x=0;
for(int a=0; a<4; a++)
{
    for(int b=0; b<a; b++)
    {
        x++;
    }
}

System.out.println("x: " + x);
```

Write a method that takes an array of doubles & returns the smallest element. If the array is empty, return Double.NaN. Include documentation comments.

Consider the Bag implemented using an array to hold data. What happens if the **insert** method is activated, and the array is already full? What is required to allow this to work even if the array had a weird length of zero?

Convert each time formula to the best possible big-O notation. Do not include any spurious constants in your big-O answer. Numerically order the entries from fastest to slowest.

Time required	Big O	Order
n		
5n		
1+2+3+...+n		
n!		
$4n^2 + 6n^3$		

In big O notation indicate the performance of the following operations of IntArrayBag:

```
// returns the current array capacity
// assume there is currently enough capacity
add(int item)

// returns the current array capacity
// assume there is currently NOT enough capacity
add(int item)

// count the number of times item is in the bag
countOccurrences(int item)

// get the total number of items in the bag
size()

// make a new bag identical to the combination of b1 & b2
union(IntArrayBag b1, IntArrayBag b2)
```

Briefly describe the purpose of a **try/catch** block.

Give an example of a method that throws an Exception.

What 2 things must a class have in order to be cloneable?

Give an example of a clone method: