

17. Refer to the doSomething method:

```
// precondition  
public static void doSomething(ArrayList<SomeType> list, int i, int j)  
{  
    SomeType temp = list.get(i);  
    list.set(i, list.get(j));  
    list.set(j, temp);  
}
```

Which best describes the *postcondition* for `doSomething`?

- (A) Removes from `list` the objects indexed at `i` and `j`.
- (B) Replaces in `list` the object indexed at `i` with the object indexed at `j`.
- (C) Replaces in `list` the object indexed at `j` with the object indexed at `i`.
- (D) Replaces in `list` the objects indexed at `i` and `j` with `temp`.
- (E) Interchanges in `list` the objects indexed at `i` and `j`.

18. Consider the `NegativeReal` class below, which defines a negative real number object.

```
public class NegativeReal
{
    private Double myNegReal;

    //constructor. Creates a NegativeReal object whose value is num.
    //Precondition: num < 0.
    public NegativeReal(double num)
    { /* implementation not shown */ }

    //Postcondition: Returns the value of this NegativeReal.
    public double getValue()
    { /* implementation not shown */ }

    //Postcondition: Returns this NegativeReal rounded to the nearest integer.
    public int getRounded()
    { /* implementation */ }
}
```

Here are some rounding examples:

<u>Negative real number</u>	<u>Rounded to nearest integer</u>
-3.5	-4
-8.97	-9
-5.0	-5
-2.487	-2
-0.2	0

Which */* implementation */* of `getRounded` produces the desired postcondition?

- (A) `return (int) (getValue() - 0.5);`
- (B) `return (int) (getValue() + 0.5);`
- (C) `return (int) getValue();`
- (D) `return (double) (getValue() - 0.5);`
- (E) `return (double) getValue();`

19. Consider the following method.

```
public static void whatsIt(int n)
{
    if (n > 10)
        whatsIt(n / 10);
    System.out.print(n % 10);
}
```

What will be output as a result of the method call `whatsIt(347)`?

- (A) 74
- (B) 47
- (C) 734
- (D) 743
- (E) 347

20. A large list of numbers is to be sorted into ascending order. Assuming that a “data movement” is a swap or reassignment of an element, which of the following is a *true* statement?
- (A) If the array is initially sorted in descending order, then insertion sort will be more efficient than selection sort.
 - (B) The number of comparisons for selection sort is independent of the initial arrangement of elements.
 - (C) The number of comparisons for insertion sort is independent of the initial arrangement of elements.
 - (D) The number of data movements in selection sort depends on the initial arrangement of elements.
 - (E) The number of data movements in insertion sort is independent of the initial arrangement of elements.