Consider a main method in a class that is included in the same package as class `GraphNode` and class `DirectedGraphNode` from the Supplementary Code. Assume that the main method contains the following code:

```java
GraphNode n1 = new GraphNode("123a");
GraphNode n2 = new DirectedGraphNode("a1", "b2", "c3");
DirectedGraphNode n3 = new DirectedGraphNode("d4", "e5", "f6");
GraphNode n4 = n2;
GraphNode n5 = n3;
```

Part I.

1. What does `System.out.println(n2)` print?

   (a) This graph node is at: 123a
   (b) This directed graph node is at: null
   (c) Nothing because the code does not compile.
   (d) Nothing because the code compiles but does not run.

2. What does `System.out.println(n4 == n2);` print?

   (a) true because they are aliases for an object with a single address.
   (b) true because at their instance variables have the same values.
   (c) false because they have different memory addresses.
   (d) false because at least one of their instance variables have different values.

3. What does `System.out.println(n2.address);` print?

   (a) 123a
   (b) a1
   (c) d4
   (d) null
   (e) none of the above
4. What does `System.out.println(n3.address);` print?

(a) 123a  
(b) a1  
(c) d4  
(d) null  
(e) none of the above

5. What does `System.out.println(n1.getAddress());` print?

(a) 123a  
(b) a1  
(c) d4  
(d) null  
(e) none of the above

6. What does `System.out.println(n2.getAddress());` print?

(a) 123a  
(b) a1  
(c) d4  
(d) null  
(e) none of the above
7. What does `System.out.println(n4.getAddress())` print?
   (a) 123a
   (b) a1
   (c) d4
   (d) null
   (e) none of the above

8. What does `System.out.println(n1.getNumNodes())` print?
   (a) 0
   (b) 1
   (c) 3
   (d) 4
   (e) none of the above

9. What does `System.out.println(n2.getNumNodes())` print?
   (a) 0
   (b) 1
   (c) 2
   (d) 3
   (e) none of the above
10. What does `System.out.println(n3.getNumNodes());` print?

(a) 0  
(b) 2  
(c) 3  
(d) 5  
(e) none of the above

11. What does `n1.printDirectedGN();` print?

(a) something  
(b) Nothing because it does not compile.  
(c) Nothing because it compiles but does not run.

12. What does `n2.printDirectedGN();` print?

(a) This is a node.  
(b) Nothing because it does not compile.  
(c) Nothing because it compiles but does not run.

13. What does `((DirectedGraphNode)n2).printDirectedGN();` print?

(a) something.  
(b) Nothing because it does not compile.  
(c) Nothing because it compiles but does not run.
For questions 14, 15, and 16, consider two functions in class MyClass with the following signatures:

```java
public static void method1(GraphNode n){...}
```
```java
public static DirectedGraphNode method2(DirectedGraphNode n){...}
```
```java
public static GraphNode method3(DirectedGraphNode n){...}
```

14. Which of the following is true about `MyClass.method1(n3);`?
   (a) It will compile and run.
   (b) It will compile, but not run.
   (c) It will not compile.

15. Which of the following is true about `MyClass.method2(n1);`?
   (a) It will compile and run.
   (b) It will compile, but not run.
   (c) It will not compile.

16. Which of the following is true about `DirectedGraphNode obj = MyClass.method3(n3);`?
   (a) It will compile and run.
   (b) It will compile, but not run.
   (c) It will not compile.
17. Write code for a class called DGNodeFactory that extends NodeFactory so that the following lines in the main method will compile and run and produce the specified output. Your DGNodeFactory should contain three variables that are of type String, called address, previous, and next respectively.

```java
public class DGNodeFactory extends NodeFactory{
    String address = "1";
    String previous;
    String next = "1a";

    public GraphNode getNewDGNode(){
        DirectedGraphNode gn = new DirectedGraphNode(previous, next, address);
        System.out.println("This is a directed graph node between " + previous + " and " + next);
        previous = address;
        address = address + "a";
        next = address + "a";
        return gn;
    }
}
```

Note that each successive node is located at an address that contains one more “a” than the previous address.

One possible solution:
Part I.

1. What does `System.out.println(n2)` print?
   (a) This graph node is at: 123a
   (b) This directed graph node is at: null
   (c) Nothing because the code does not compile.
   (d) Nothing because the code compiles but does not run.

2. What does `System.out.println(n4 == n2)` print?
   (a) true because they are aliases for an object with a single address.
   (b) true because at their instance variables have the same values.
   (c) false because they have different memory addresses.
   (d) false because at least one of their instance variables have different values.

3. What does `System.out.println(n3.address)` print?
   (a) 123a
   (b) a1
   (c) d4
   (d) null
   (e) none of the above
4. What does `System.out.println(n2.address);` print?

(a) 123a
(b) a1
(c) d4
(d) null
(e) none of the above

5. What does `System.out.println(n1.getAddress());` print?

(a) 123a
(b) a1
(c) d4
(d) null
(e) none of the above

6. What does `System.out.println(n2.getAddress());` print?

(a) 123a
(b) a1
(c) d4
(d) null
(e) none of the above
7. What does `System.out.println(n4.getAddress())` print?
   (a) 123a
   (b) a1
   (c) d4
   (d) null
   (e) none of the above

8. What does `System.out.println(n1.getNumNodes());` print?
   (a) 0
   (b) 1
   (c) 3
   (d) 4
   (e) none of the above

9. What does `System.out.println(n2.getNumNodes());` print?
   (a) 0
   (b) 1
   (c) 2
   (d) 3
   (e) none of the above
Don’t forget to record your answers on the front page, except for question #17.

10. What does `System.out.println(n3.getNumNodes());` print?
   (a) 0  
   (b) 2  
   (c) 3  
   (d) 5  
   (e) none of the above

11. What does `n1.printDirectedGN();` print?
   (a) This is a node.  
   (b) Nothing because it does not compile.  
   (c) Nothing because it compiles but does not run.

12. What does `n2.printDirectedGN();` print?
   (a) This is a node.  
   (b) Nothing because it does not compile.  
   (c) Nothing because it compiles but does not run.

13. What does `((DirectedGraphNode)n2).printDirectedGN();` print?
   (a) Something  
   (b) Nothing because it does not compile.  
   (c) Nothing because it compiles but does not run.
Don’t forget to record your answers on the front page, except for question #17.

For questions 14, 15, and 16, consider two functions in class MyClass with the following signatures:

public static void method1(GraphNode n){...}

public static DirectedGraphNode method2(DirectedGraphNode n){...}

public static GraphNode method3(DirectedGraphNode n){...}

14. Which of the following is true about MyClass.method1(n3)?
   (a) It will not compile.
   (b) It will compile, but not run.
   (c) It will compile and run.

15. Which of the following is true about MyClass.method2(n1)?
   (a) It will not compile.
   (b) It will compile, but not run.
   (c) It will compile and run.

16. Which of the following is true about DirectedGraphNode obj = MyClass.method3(n3)?
   (a) It will not compile.
   (b) It will compile, but not run.
   (c) It will compile and run.
17. Write code for a class called `DGNodeFactory` that extends `NodeFactory` so that the following lines in the `main` method will compile and run and produce the specified output. Your `DGNodeFactory` should contain three variables that are of type `String`, called `address`, `previous`, and `next` respectively.

```java
dgnodefactory fac = new dgnodafactory();
graphnode n = fac.getnewdgnode();
    //prints: This is a directed graph node between null and 1a
graphnode nn = fac.getnewdgnode();
    //prints: This is a directed graph node between 1 and 1aa
graphnode nnn = fac.getnewdgnode();
    //prints: This is a directed graph node between 1a and 1aaa
```

Note that each successive node is located at an address that contains one more “a” than the previous address.

One possible solution:

```java
public class DGNodeFactory extends NodeFactory{
    String address = "1";
    String previous;
    String next = "1a";

    public GraphNode getNewDGNode(){
        DirectedGraphNode gn = new DirectedGraphNode(previous, next, address);
        System.out.println("This is a directed graph node between " + previous + " and " + next);
        previous = address;
        address = address + "a";
        next = address + "a";
        return gn;
    }
}
```