Explain what happens when the program is compiled and then executed. It might be best to draw what is happening during execution in the form of a memory model. Of interest are for example:

Ans: When the program is compiled and executed, the first object is created named aHelloWorld1 with the constructor having no parameters. This object has aString = “hello”, static bString = “hello” and incremented static aInt = 1. Then a second object named aHelloWorld2 is created with the constructor containing one parameter. This object changes the value of the static aInt = 4. And has aString = null because the aString value is not initialized. And bString = “hello” since it is static and was already updated. Then an anonymous object is created which increments the aInt value to 5. Now, the setBoth method is called on aHelloWorld1 with aHelloWorld2 as the parameter. In this method, the aInt value changes to 42 and the setState method with parameter 84 is called on the aHelloWorld parameter. Now, in the setState method, it changes the static aInt value to 84. Then, the setStrings is called on aHelloWorld1, this sets the aString value to hello0123456789 and static bString to hello0123456789, then the setStrings is called on aHelloWorld2, and that changes the aString value to 0123456789 since it was an empty string and static bString to hello01234567890123456789.

- **How many class variables do exist, for how long?**

Ans: There are two class variables, which are - aInt and bString, and they exist permanently till the end.

- **How many instance variables do exist, for how long?**

Ans: There is just one instance variable named aString and exists till the object in which it is stored exists.

- **How and when are these methods called? What is the receiving object?**

Ans: The methods are called in the main function.

```
public static void main(String args[] ) {

        HelloWorld aHelloWorld1 = new HelloWorld();   //here the aHelloworld1 obj is created and constructor is called

        HelloWorld aHelloWorld2 = new HelloWorld(4); //same here just that over here it is called with the parameter aint as 4

        new HelloWorld();  //constructor called without any new obj being created

        aHelloWorld1.setBoth(aHelloWorld2);  //setBoth method called with aHelloworld2 as parameter.
```

```
                aHelloWorld.setState(aInt * 2 );  //setState method called with aint*2 passed as a parameter.

    aHelloWorld1.setStrings();   //setStrings method called

    aHelloWorld2.setStrings();  //setString method called for another aHelloWorld2

    System.out.println( aHelloWorld1);  //toString method called by default

    System.out.println( aHelloWorld2);  //toString method called by default

    System.out.println( getCounter());  //toString method called by default
```

• Explain the call sequence of the constructor calls.

Ans:

HelloWorld aHelloWorld1 = new HelloWorld();   - constructor called and stored in aHelloWorld1 object.

HelloWorld aHelloWorld2 = new HelloWorld(4); -constructor called and stored in aHelloWorld2 object.

new HelloWorld(); -constructor called but it is not stored in an object.

• Is it possible to change all public methods to private?

Ans: In this particular code yes, except the toString().