Q1

Output:

```
b
c
a
100
107
207
214
421
2561
```

[0.25x9=2.25]

Note 1: Give marks for **b** only when it's printed before **c**.

Note 2: Give marks for **a** only when it's printed before **100.**

[{0.25+0.25}x8=4]: 0.25 marks for identifying the correct print statement. 0.25 marks for the explanations given.

1) **b:** It gets printed due to the *System.out.println("b")* statement in the static block of the quiz class. A Static block gets executed even before the main function.
2) **c:** It gets printed due to the *System.out.println(“c")* statement. This is the first print statement of the main function and prints ‘c’.
3) **a:** It gets printed due to the *System.out.println(“a")* statement in the static block of Scaler class. Whenever a class is used for the first time, its static block gets executed. The first *System.out.println(Scaler.i)* statement was the first to use the Scaler class.
4) **100:** It gets printed due to the first *System.out.println(Scaler.i)* statement. ‘i’ already got initialized to 100 in the static block execution in the previous step.
5) **107:** It gets printed due to the first *System.out.println(new Scaler().j)* statement. A initialization block gets executed before the default constructor. Hence, ‘j’ gets initialized to ‘i+7’.
6) **207:** It gets printed due to the second *System.out.println(Scaler.i)* statement. ‘i’ is 207 because it got increment by ‘j’ in the initialization block in the previous step.
7) **214:** It gets printed due to the second *System.out.println(new Scaler().j)* statement. The initialization block gets called again, but ‘j’ gets initialized differently because of the updated ‘i’ in the previous step.
8) **421:** It gets printed due to the third *System.out.println(Scaler.i)* statement. ‘i’ already got updated in the previous execution of the initialization block.

0.25 marks for identifying the correct print statement. 1 mark for the explanations given.

2561: It gets printed due to the *System.out.println(Scaler.update_i(new Scaler()).i)* statement, where the static update_i function is being called to return the value of updated ‘i’ for printing. In this function, a new Scaler object is being passed, which means ‘i’ already gets incremented with the new initialization value of ‘j’ in the initialization block, resulting in i=849, with a.j=428. In side this function both these values get added to update ‘i’ as 1277. The function returns reference of another new object, calling the initialization block again. This results in i=2561, with new initialization value of j as 1284.

Q2

Output

```
hey
hi
bye
hi again
bye again
```

[0.25*5=1.25]

Note 1: Give marks for ‘hey’ only when it’s the first output.

- **hey:** static block gets executed first, which initializes static variable ‘i’ as ‘hey’ and calls the second main function to print ‘i’. [0.25 marks]
- **hi & bye:** After the execution of second main function, the control goes back to the static block, which initializes the static variable ‘j’ now and calls the first main function to print ‘j’. [0.5 marks]
- **hi again & bye again:** After execution of the first main function, the control goes back to the static function, which now sets ‘ctr’ to 1. After the static block execution is over, JVM calls the first main function [the default main function having the required signature]. Since ‘ctr’ is set to 1 now, ‘j’ gets updated and the main function prints the updated ‘j’. [0.5 marks]