## Short answer

#### A

Imperative: The developer has to control every step of the function, including every variable initialization and computation. The developer has to consider not only what to do but also hot to do.

Declarative: It’s functional. The developer focuses more on what to do rather than how to do.

#### B

The meaning of declarative programming is to express what to do rather than how to do it. A very good example is the forEach() method in the Iterable interface. We assign a function to the target List and call this function on every member of the target List.

#### C Explain the difference between functional interface, functor, and closure, and give examples of each using Java 7 syntax

A closure is a functor embedded inside another class and that is capable of remembering the state of its enclosing object.

A functional interface is an interface with only one abstract method interface.

#### D Name three benefits of including functional style programming in Java

#### E

x -> x + 2\*x\*x;

(x, y) -> y – x + Math.pow(x, y)

(x, y, z) -> z – (x + y)

#### F

1. No parameter; Free variables: s, t
2. Parameters: u, v; Free variables: a, b
3. Parameters: s, t; Free variables: ignoreCase

#### G

Prediction

#### H

1. s -> System.out.println(s)
2. object::instanceMethod, because out is an object in class System.

#### J

1. Supplier supplier = () -> Math.random();
2. System.out.println(supplier.get());