Examen de Certificación V

1. What is the result of the following class?

```
1: import java.util.function.*;
2:
3: public class Panda {
4: int age;
5: public static void main(String[] args) {
6: Panda p1 = new Panda();
7: p1.age = 1;
8: check(p1, p -> p.age < 5);
9: }
10: private static void check(Panda panda,
11: Predicate<Panda> pred) {
12: String result =
13: pred.test(panda) ? "match" : "not match";
14: System.out.print(result);
15: } }
```

A. match
B. not match
C. Compiler error on line 8.
D. Compiler error on lines 10 and 11.
E. Compiler error on lines 12 and 13.
F. A runtime exception is thrown.

2. Which of the following will compile when filling in the blank? (Choose all that apply.)

```
List list = List.of(1, 2, 3);
Set set = Set.of(1, 2, 3);
Map map = Map.of(1, 2, 3, 4);
___________.forEach(x -> System.out.println(x));
```

A. list
B. set
C. map
D. map.keys()
E. map.keySet()
F. map.values()
G. map.valueSet()

3. How many lines does this code output?

```
Set<String> set = Set.of("mickey", "minnie");
List<String> list = new ArrayList<>(set);
set.forEach(s -> System.out.println(s));
list.forEach(s -> System.out.println(s));
```

A. 0
B. 2
C. 4
D. The code does not compile.
E. A runtime exception is thrown.

4. What is the output of the following code?

```
List<String> cats = new ArrayList<>();
cats.add("leo");
cats.add("Olivia");
cats.sort((c1, c2) -> -c1.compareTo(c2)); // line X
System.out.println(cats);
```

A. [leo, Olivia]
B. [Olivia, leo]
C. The code does not compile because of line X.
D. The code does not compile for another reason.
E. A runtime exception is thrown.

5. Which is true of the following code?

```
int length = 3;
for (int i = 0; i<3; i++) {
      if (i%2 == 0) {
            Supplier<Integer> supplier = () -> length; // A
            System.out.println(supplier.get()); // B
      } else {
            int j = i;
            Supplier<Integer> supplier = () -> j; // C
            System.out.println(supplier.get()); // D
      }
}
```

A. The first compiler error is on line A.
B. The first compiler error is on line B.
C. The first compiler error is on line C.
D. The first compiler error is on line D.
E. The code compiles successfully.

6. What is the output of the following code?

```
Set<String> cats = new HashSet<>();
cats.add("leo");
cats.add("Olivia");
cats.sort((c1, c2) -> -c1.compareTo(c2)); // line X
System.out.println(cats);
```

A. [leo, Olivia]
B. [Olivia, leo]
C. The code does not compile because of line X.
D. The code does not compile for another reason.
E. A runtime exception is thrown.

7. How many lines does this code output?

```
Set<String> s = Set.of("mickey", "minnie");
List<String> x = new ArrayList<>(s);
s.forEach(s -> System.out.println(s));
x.forEach(x -> System.out.println(x));
```

A. 0
B. 2
C. 4
D. The code does not compile.
E. A runtime exception is thrown.

8. Given: Which of the following lambda expressions can be passed to a function of Predicate<String> type? (Choose all that apply.)

A. s -> s.isEmpty()
B. s --> s.isEmpty()
C. (String s) -> s.isEmpty()
D. (String s) --> s.isEmpty()
E. (StringBuilder s) -> s.isEmpty()
F. (StringBuilder s) --> s.isEmpty()

9. Which lambda can replace the MyTest class to return the same value? (Choose all that apply)

```
interface Sample {
    String change(int d);
}
class MyTest implements Sample {
    public String change(int s) {
    return "Hello";
   }
}
```

A. change((e) -> “Hello”)
B. change((e) -> {“Hello” })
C. change((e) -> { String e = “”; “Hello” });
D. change((e) -> { String e = “”; return “Hello”; });
E. change((e) -> { String e = “”; return “Hello” });
F. change((e) -> { String f = “”; return “Hello”; });

10. Which are true about the functional interface?

A. It has exactly one method and it must be abstract.
B. It has exactly one method and it may or may not be abstract.
C. It must have exactly one abstract method and may have any number of default or static methods.
D. It must have exactly one default method and may have any number of abstract or static methods.
E. It must have exactly one static method and may have any number of default or abstract methods.