Started on	Wednesday, 10 March 2021, 19:49
State	Finished
Completed on	Sunday, 14 March 2021, 17:14
Time taken	3 days 21 hours
Grade	**80.00** out of 150.00 (**53**%)

Question 1

Correct Mark 10.00 out of 10.00

In the buggy Java code below, is the bug caught by static checking, dynamic checking, or not at all?

```
int n = 5;
if (n) {
    System.out.println("n is " + n);
}
```

Select one:

- a. static checking
- b. dynamic checking
- c. no checking, resulting in wrong answer

Your answer is correct.

The correct answer is: static checking

Question 2

Incorrect Mark 0.00 out of 10.00

In the buggy Java code below, is the bug caught by static checking, dynamic checking, or not at all?

```
int bigNum = 200000;        // bigNum is 200,000
bigNum = bigNum * bigNum;   // bigNum should be 4 billion
```

Select one:

- a. static checking
- b. dynamic checking
- c. no checking, resulting in wrong answer

Your answer is incorrect.

The correct answer is: no checking, resulting in wrong answer

Question 3

Incorrect Mark 0.00 out of 10.00

In the buggy Java code below, is the bug caught by static checking, dynamic checking, or not at all?

```
// the probability of an event is prob = 1/5 = 0.2
double prob = 1/5;
```

Select one:

a. static checking

b. dynamic checking

c. no checking, resulting in wrong answer

Your answer is incorrect.

The correct answer is: no checking, resulting in wrong answer

Question 4

Correct Mark 10.00 out of 10.00

In the buggy Java code below, is the bug caught by static checking, dynamic checking, or not at all?

```
int sum = 0;
int n = 0;
int average = sum/n;
```

Select one:

a. static checking

b. dynamic checking

c. no checking, resulting in wrong answer

Your answer is correct.

The correct answer is: dynamic checking

Question 5

Incorrect Mark 0.00 out of 10.00

In the buggy Java code below, is the bug caught by static checking, dynamic checking, or not at all?

```
double sum = 7;
double n = 0;
double average = sum/n;
```

Select one:

a. static checking

b. dynamic checking

c. no checking, resulting in wrong answer

Your answer is incorrect.

The correct answer is: no checking, resulting in wrong answer

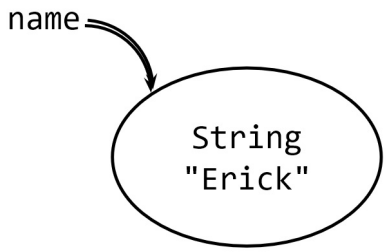
Question 6

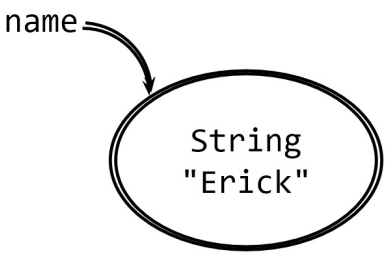
Correct Mark 10.00 out of 10.00

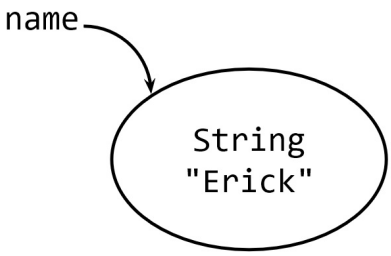
Which is the correct snapshot diagram for:

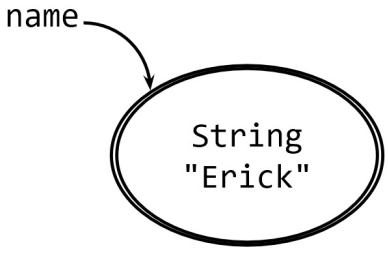
final String name = "Erick";

Select one:









Your answer is correct.

The correct answer is:

Question 7

Incorrect Mark 0.00 out of 10.00

Choose the **incorrect** statement:

Select one:

a. String is an immutable type.

b. StringBuilder is a mutable type.

c. final variable cannot be reassigned.

d. object pointed by final variable cannot be mutated.

e. List is a mutable type.

Your answer is incorrect.

The correct answer is: object pointed by final variable cannot be mutated.

Question 8

Incorrect Mark 0.00 out of 20.00

When you try to reassign a final variable, Java compiler will produce a compile error.

Therefore, final provides you immutable reference ✗ for immutable values ✗ .

Question 9

Incorrect Mark 0.00 out of 10.00

Rewrite the variable declaration below using Lists instead of arrays:

```
char[][] matrix;
```

Answer: List matrix = new ArrayList<>() ✗

The correct answer is: List<List<Character>> matrix ;

Question 10

Correct Mark 10.00 out of 10.00

Given a code:

```
List<Integer> list1 = new ArrayList<>();
list1.add(100);
list1.add(200);
final List<Integer> list2 = list1;
list1.add(300);
```

If we add a line of code below:

```
list2 = list1;
```

choose the **correct** statement:

Select one:

- a. there will be an error, detected by static checking.
- b. there will be an error, detected by dynamic checking.
- c. there is no error.

Your answer is correct.

The correct answer is: there will be an error, detected by static checking.

Question 11

Correct Mark 10.00 out of 10.00

Given a code:

```
List<Integer> list1 = new ArrayList<>();
list1.add(100);
list1.add(200);
final List<Integer> list2 = list1;
list1.add(300);
```

If we add a line of code below:

```
list1.set(1, 400);
```

choose the **correct** statement:

Select one:

- a. there will be an error, detected by static checking.
- b. there will be an error, detected by dynamic checking.
- c. there is no error.

Your answer is correct.

The correct answer is: there is no error.

Question 12

Correct Mark 10.00 out of 10.00

Given a code:

```
List<Integer> list1 = new ArrayList<>();
list1.add(100);
list1.add(200);
final List<Integer> list2 = list1;
list1.add(300);
```

If we add a line of code below:

```
list2.set(1, 400);
```

choose the **correct** statement:

Select one:

- a. there will be an error, detected by static checking.
- b. there will be an error, detected by dynamic checking.
- c. there is no error.

Your answer is correct.

The correct answer is: there is no error.

Question 13

Correct Mark 10.00 out of 10.00

Given a code:

```
List<Integer> list1 = new ArrayList<>();
list1.add(100);
list1.add(200);
final List<Integer> list2 = list1;
list1.add(300);
```

If we add a line of code below:

```
list2.set(3, 400);
```

choose the **correct** statement:

Select one:

- a. there will be an error, detected by static checking.
- b. there will be an error, detected by dynamic checking.
- c. there is no error.

Your answer is correct.

The correct answer is: there will be an error, detected by dynamic checking.

Question 14

Correct Mark 10.00 out of 10.00

Create a map named hostel with integer keys and string values, to store room number and tenant name pairs.

Then, add a key-value pair for a tenant named Alice in room number 777.

Select one:

- Map<Integer, String> hostel = new HashMap<>();
  hostel.add(777, "Alice");
- Map<Integer, String> hostel = new HashMap<>();
  hostel.put(777, "Alice");
- Map<String, Integer> hostel = new HashMap<>();
  hostel.add("Alice", 777);
- Map<String, Integer> hostel = new HashMap<>();
  hostel.put("Alice", 777);
- Map<String, int> hostel = new HashMap<>();
  hostel.add("Alice", 777);
- Map<String, int> hostel = new HashMap<>();
  hostel.put("Alice", 777);

Your answer is correct.
The correct answer is:

Map<Integer, String> hostel = new HashMap<>();

hostel.put(777, "Alice");

Finish review

Lab 2 Videos Jump to... Lab Exercise 2.1 Max Stret