



[Home](#) - [My courses](#) - [CPT204\(S2\)](#) - [Sections](#) - [Week 6 : 5-9 April — Exception, Deep Copy, Copy Constructor](#) - [Lecture Quiz 6](#)

Started on Thursday, 8 April 2021, 15:49

State Finished

Completed on Thursday, 8 April 2021, 17:07

Time taken 1 hour 18 mins

Grade **70.00** out of 130.00 (54%)

Question 1

Correct

Mark 20.00 out of 20.00

Which of the following **cannot** be null?

Select one or more:

- ☒ `char c;`
- ☐ `static final String str;`
- ☐ `int[] arr;`
- ☐ `Double d;`
- ☐ `final BackAccount myBankAccount;`
- ☐ `String name;`
- ☒ `double d;`

Your answer is correct.

The correct answers are: `char c;`,
`double d;`

Question 2

Correct

Mark 10.00 out of 10.00

Given the following code :

```
public static String nope() {  
    return null;           // (1)  
}  
  
public static void main(String[] args) {  
    String a = nope();      // (2)  
    String b = null;        // (3)  
    if (a.length() > 0) {   // (4)  
        b = a;              // (5)  
    }  
    return b;              // (6)  
}
```

Which line contains a static error?

Which line contains a static error :

Select one:

- ☐ (1)
- ☐ (2)
- ☐ (3)
- ☐ (4)
- ☐ (5)
- ☒ (6)

Your answer is correct.

The correct answer is: (6)

Question 3

Correct

Mark 10.00 out of 10.00

Given the same code from Question 2 above :

```
public static String nope() {  
    return null;           // (1)  
}  
  
public static void main(String[] args) {  
    String a = nope();      // (2)  
    String b = null;        // (3)  
    if (a.length() > 0) {   // (4)  
        b = a;              // (5)  
    }  
    return b;              // (6)  
}
```

Suppose you have commented out the line causing the static error in Question 2.

Now, which line contains a dynamic error ?

Select one:

- ☐ (1)
- ☐ (2)
- ☐ (3)
- ☒ (4)
- ☐ (5)
- ☐ (6)

Your answer is correct.

The correct answer is: (4)

Question 4

Incorrect

Mark 0.00 out of 10.00

Suppose we're building a robot and we want to specify the method

```
public static List<Point> findPath(Point initial, Point goal)
```

which is responsible for path-finding: determining a sequence of `Points` that the robot should move through to navigate from `initial` to `goal`, past any obstacles that might be in the way.

In the postcondition, we say that `findPath` will search for paths only up to a bounded length (set elsewhere), and that ***it will throw an exception if it fails to find one.***

Which exception is the best exception and its type to create, according to Lecture 6?

Select one:

- ☐ a. a checked `PathNotFoundException`
- ☒ b. an unchecked `PathNotFoundException`
- ☐ c. a checked `NoPathException`
- ☐ d. an unchecked `NoPathException`

Your answer is incorrect.

The correct answer is: a checked `PathNotFoundException`

Question 5

Incorrect

Mark 0.00 out of 10.00

Suppose we define a checked exception for the method `findPath`.

What will we choose as our superclass?

Select one:

- ☐ a. `Exception`
- ☒ b. `Throwable`
- ☐ c. `Error`
- ☐ d. `RuntimeException`

Your answer is incorrect.

The correct answer is: `Exception`

Question 6

Incorrect

Mark 0.00 out of 10.00

Suppose we define an unchecked exception for the method `findPath`.

What will we choose as our superclass?

Select one:

- ☒ a. `Exception`

- ☐ b. Throwable
- ☐ c. Error
- ☐ d. RuntimeException

Your answer is incorrect.

The correct answer is: RuntimeException

Question 7

Incorrect

Mark 0.00 out of 20.00

Consider this code below for analyzing some **Thing** objects:

```
static List<Thing> allTheThings;

static void analyzeEverything() {
    analyzeThings();
}

static void analyzeThings() {
    try {
        for (Thing t : allTheThings) {
            analyzeOneThing(t);
        }
    } catch (AnalysisException ae) {
        return;
    }
}

static void analyzeOneThing(Thing t) throws AnalysisException {
    // ...
    // ... maybe go past the end of a list
    // ...
}
```

Note that **IndexOutOfBoundsException**, **NullPointerException**, and **OutOfMemoryError** are unchecked exceptions and **AnalysisException** is a checked exception.

Which exception could be thrown by a call to **analyzeEverything**?

Select one or more:

- ☒ **AnalysisException**
- ☐ IndexOutOfBoundsException
- ☐ NullPointerException
- ☐ OutOfMemoryError

Your answer is incorrect.

The correct answers are: IndexOutOfBoundsException, NullPointerException, OutOfMemoryError

Question 8

Partially correct

Mark 10.00 out of 20.00

If we want to construct a different object with the same values as the input object, we use a/an

other

✗ that performs a/an


deep copy  instead of a shallow copy.

Question 9**Correct****Mark 10.00 out of 10.00**

Write one line of Java code that throws an `IllegalArgumentException` object with a message "n must not be even" to complete the if statement below :

```
if (n % 2 == 0) {  
    // your code here  
}
```

Do not forget to end it with a semicolon.


Answer: throw new IllegalArgumentException("n must not be even"); 

The correct answer is: throw new IllegalArgumentException("n must not be even");

Question 10**Correct****Mark 10.00 out of 10.00**

When we throw an `IllegalArgumentException` object within a method, that method must advertise it in the method signature.

Select one:

- ☐ True
- ☒ False 

The correct answer is 'False'.

Finish review

◀ Lab 6 Recording

Jump to...

Lab Exercise 6.1 Vehicle CONS