

# Final

**Due** Aug 17 at 11:59pm

**Points** 60

**Questions** 60

**Available** Aug 12 at 12am - Aug 17 at 11:59pm

**Time Limit** 120 Minutes

## Attempt History

	Attempt	Time	Score
LATEST	<a href="#">Attempt 1</a>	59 minutes	59 out of 60

Score for this quiz: **59** out of 60

Submitted Aug 15 at 11:26am

This attempt took 59 minutes.

### Question 1

1 / 1 pts

What exactly is a JFrame?

☐ A light-weight Swing component that holds other Swing components

☐ The parent class of all Swing components

☒ A heavy-weight Swing component used as a top-level container for other components

☐ Swing component for displaying simple text

Correct!

### Question 2

1 / 1 pts

What is the main difference between a JTextField and a JPasswordField?

**Correct!**

- ☒ A JPasswordField will mask input while a JTextField will not
- ☐ There is no difference
- ☐ A JTextField can accept user input but a JPasswordField is only for displaying information
- ☐ A JPasswordField will not allow you to retrieve the typed in value

### Question 3

1 / 1 pts

In Java, what does it mean to make your code “thread safe”?

- ☐ It means to lock down the ability to create new threads
- ☐ It means ensuring that no exceptions happen during the execution of the thread
- ☒ It means to allow different threads access to the same resources without causing problems
- ☐ It means limiting how many threads can be created

**Correct!**

### Question 4

1 / 1 pts

Which of the following statements is true regarding threads in Java?

- ☐ Contrary to what you might think, a multi-threaded Java application actually cannot do multiple tasks at the same time

Correct!



The number of threads that can be created depends on many factors but in many cases there is no hard limit



You would only use a thread pool in a small application that does not need very many threads



Unless you go out of your way in the code to start a different thread, a Java process will have exactly one thread

### Question 5

1 / 1 pts

What is the main difference between an abstract class and an interface?



An interface can have static methods, an abstract class cannot



Only an interface can define methods without a body



Polymorphism can only be achieved using an abstract class



An interface cannot have a constructor or hold state, but an abstract class can

Correct!

### Question 6

0 / 1 pts

Which of the following, if present inside an abstract class, will cause it to not compile?



static constructor



inner classes

Correct Answer

☒ private instance variables☐ abstract methods**Question 7****1 / 1 pts**

Which of these statements regarding the Java language is true?

- ☐ It is both statically and dynamically typed
- ☐ It is a dynamically typed language
- ☐ It is neither statically nor dynamically typed
- ☒ It is a statically typed language

**Correct!****Question 8****1 / 1 pts**

Which of the following is a valid reason to use generics in Java?

- ☐ To ensure type safety during runtime
- ☐ To speed up compilation
- ☐ To encourage type casting
- ☒ To take advantage of code re-use

**Correct!****Question 9****1 / 1 pts**

What is the difference between a JDialog and a JFrame?

- ☐ There is no functional difference
- ☐ A JFrame is simply a subclass of JDialog with a bit more functionality added
- ☒ A JDialog will not have maximize and minimize buttons and is not considered an application window
- ☐ A JDialog cannot have components added to it like a JFrame can

Correct!

### Question 10

1 / 1 pts

What is the difference between the “super” keyword and the “this” keyword?

- ☐ They actually both refer to the same thing, the current object
- ☐ “super” is a keyword that can be put onto a method to prevent it from being overridden
- ☒ “super” refers to the parent object and “this” refers to the current object
- ☐ You can have a variable named “super” but you cannot have a variable named “this”

Correct!

### Question 11

1 / 1 pts

Which of the following is true regarding the EDT?

Correct!

- ☒ It is a dedicated thread from which Swing object interactions should take place
- ☐ It is fine to run both short and long running tasks from it
- ☐ It should rarely be used when working with Swing
- ☐ It is a thread pool that can be tapped into to make your Swing app faster

## Question 12

1 / 1 pts

What exactly is an Error?

- ☐ Java does not use them, it only uses exceptions
- ☒ A very serious problem that should not be caught in a try-catch
- ☐ Every exceptions has a field of type Error with info about the exception
- ☐ It is the superclass for all types of exceptions

Correct!

## Question 13

1 / 1 pts

Which of the following code snippets is the correct way to create a lambda with 1 argument that returns an int?

- ☐ (arg) => 50;
- ☐ (arg) > 50;
- ☐ (arg) = 50;
- ☒ (arg) -> 50;

Correct!

## Question 14

1 / 1 pts

What is the difference between a non-static inner class and a static inner class?

Correct!



A non-static inner class can access the enclosing class's instance variables, a static inner class cannot



Only a static inner class can reside inside an abstract class



A non-static inner class can have its own custom constructor, a static inner class cannot



A non-static inner class can be extended by a different class, but a static inner class cannot be

## Question 15

1 / 1 pts

What is the main difference between a JTextField and a JLabel?

Correct!



A JLabel can be clicked but a JTextField can't be clicked



A JTextField is for accepting variable input from the user and a JLabel can't be typed into



A JLabel can be added to a JOptionPane but a JTextField can't be



A JTextField is considered to be a container component in Swing but a JLabel is not

### Question 16

1 / 1 pts

Which of the following is proper syntax for catching multiple exceptions in one catch block?

- ☐ `catch (ClassCastException e1, NullPointerException e2) {}`
- ☐ `catch (ClassCastException,NullPointerException e) {}`
- ☒ `catch (ClassCastException|NullPointerException e) {}`
- ☐ `catch (ClassCastException e) {} catch(NullPointerException e) {}`

Correct!

### Question 17

1 / 1 pts

Which of the following is a true statement?

- ☐ Compiled Java code is not platform independent
- ☐ Starting up the JVM does not have any performance costs
- ☒ You would use the JDK over the JRE if you intended to compile .java files
- ☐ It is rare for a Java release to be backwards compatible with previous releases

Correct!

### Question 18

1 / 1 pts

Say you have a class called `FileProcessor` which is abstract and has a single abstract method declared. The class `CSVProcessor` extends `FileProcessor`. Given this scenario, which of the following statements is true?



Correct!



CSVProcessor does not need to do anything beyond “extends FileProcessor” in order to compile



CSVProcessor needs to be marked as abstract or override the abstract method in FileProcessor in order to compile



The only way it will compile is if CSVProcessor is also marked as abstract



The only way it will compile is if CSVProcessor overrides the abstract method

### Question 19

1 / 1 pts

Which of the following statements is true?



A lambda must always return a value within the method implementation



A lambda must be created when implementing single method interfaces



A lambda is simply a way of creating concise anonymous inner classes



A lambda cannot create another lambda

Correct!

### Question 20

1 / 1 pts

What is the difference between an “Integer” variable and an “int” variable?



They have different min and max values



“Integer” is nullable

Correct!

- ☐ There is no difference
- ☐ “Integer” is deprecated and we should always use “int” going forward

### Question 21

1 / 1 pts

What is type erasure?

- ☐ The ability to pass “Void” as a generic type
- ☐ Refers to the fact that all reference types in Java are descended from Object
- ☒ The process of enforcing type constraints only at compile time
- ☐ When generic types are overloaded

Correct!

### Question 22

1 / 1 pts

What exactly is JOptionPane?

- ☐ A Swing component for showing drop down values
- ☐ A type of dialog that has tabs at the top
- ☐ A Swing component for doing radio buttons
- ☒ A utility class for creating common dialog formats

Correct!

### Question 23

1 / 1 pts

What is correct syntax for adding an action listener to a JButton?

- ☐ myButton.addActionListener(System.out.println("clicked"));
- ☐ myButton.clicked(e -> System.out.println("clicked"));
- ☒ myButton.addActionListener(e -> System.out.println("clicked"));
- ☐ myButton.addActionListener(() -> System.out.println("clicked"));

Correct!

## Question 24

1 / 1 pts

What does the synchronized keyword do when put onto a method?

- ☐ It allows the Java system to guarantee the code in that method is thread safe
- ☐ It is another way of marking a method as static
- ☒ It causes the method to run one thread at a time and other threads get backed up
- ☐ It will run the method in its own thread

Correct!

## Question 25

1 / 1 pts

What happens if you try to add a duplicate item to a HashSet?

- ☐ You will receive an exception
- ☒ The add method will return false
- ☐ It will not compile

Correct!

- ☐ It will add the duplicate item to the set

### Question 26

1 / 1 pts

In computing, what is the difference between a thread and a process?

- ☐ They are actually the exact same thing
- ☒ A thread is sometimes called a lightweight process but it is not an application
- ☐ A single thread manages multiple processes, not the other way around
- ☐ A thread can be started from inside Java but a process cannot

Correct!

### Question 27

1 / 1 pts

Under what circumstances would you use a try-with-resources?

- ☐ To catch exceptions that arise from interacting with a particular resource
- ☒ To avoid needing to close resources in a finally block
- ☐ You shouldn't, ever. It is sinful to do so.
- ☐ To allow unchecked exception handling to compile

Correct!

### Question 28

1 / 1 pts

Which of the following is correct syntax for declaring a generic method?

Correct!

- ☐ public void <K> bla() {}
- ☒ public <K> void bla() {}
- ☐ public void bla<K>() {}
- ☐ <K> public void bla() {}

### Question 29

1 / 1 pts

Which of the following statements will not compile?

- ☐ ArrayList<Boolean> a = new ArrayList<>();
- ☒ List<Boolean> a = new ArrayList<>;
- ☐ List<Boolean> a = new ArrayList<Boolean>();
- ☐ List<Boolean> a = new ArrayList<>();

Correct!

### Question 30

1 / 1 pts

What happens when a class does not specify its own constructor?

- ☒ It uses the default object constructor that does not have arguments
- ☐ It is the same as having a private constructor
- ☐ The compiler generates a constructor that has a parameter for each instance variable

Correct!



It will not compile because in order to create an object it needs to have a constructor

### Question 31

1 / 1 pts

Which of the following is correct syntax for a lambda with zero arguments that returns a boolean?

☐ (null) -> false

☐ -> false

☒ () -> false

☐ true

Correct!

### Question 32

1 / 1 pts

What does “Thread.sleep(1000);” do?

☐ Causes the current thread to die in one second

☒ Causes the current thread to pause for one second

☐ Waits one second before starting a new thread

☐ Caches a thread with a one second revival time

Correct!

### Question 33

1 / 1 pts

Which of the following statements is true about Swing?

Correct!

- ☒ It is platform independent and typically looks the same on different OS's
- ☐ It is a completely dead technology that is not supported anymore
- ☐ It is not still a part of the core JDK in the latest versions
- ☐ Nowadays it is only used to make websites

### Question 34

1 / 1 pts

What exactly is a JComponent?

Correct!

- ☒ The parent class of all Swing components
- ☐ A light-weight Swing component that holds other Swing components
- ☐ Swing component for allowing users to initiate an action with a mouse click
- ☐ A heavy-weight Swing component used as a top-level container for other components

### Question 35

1 / 1 pts

What exactly is a JLabel?

- ☐ A heavy-weight Swing component used as a top-level container for other components

Correct!

- ☐ A light-weight Swing component that holds other Swing components
- ☐ Swing component for allowing users to initiate an action with a mouse click
- ☒ Swing component for displaying simple text

### Question 36

1 / 1 pts

What exactly is a JPanel?

- ☐ Swing component for displaying simple text
- ☒ A light-weight Swing component that holds other Swing components
- ☐ A heavy-weight Swing component used as a top-level container for other components
- ☐ The parent class of all Swing components

Correct!

### Question 37

1 / 1 pts

What is an enum?

- ☒ A type of variable in Java that has an enumerated set of possible values
- ☐ A type of variable that can be assigned a null value
- ☐ A type of class that can be extended by another class
- ☐ A numeric data type in Java

Correct!



### Question 38

1 / 1 pts

What exactly is a stateless object in Java?

- ☐ An object that can cross “state lines” between two threads
- ☒ An object with no fields and which doesn't reference fields from other classes
- ☐ An object that can be serialized
- ☐ An object that cannot be instantiated

Correct!

### Question 39

1 / 1 pts

What happens if you don't specify an access modifier (private, public, or protected) on a variable?

- ☐ By default variables are marked as public if no access modifier is specified
- ☐ It will not compile
- ☒ Not specifying a modifier is actually its own access level called the “default” modifier
- ☐ It makes it so nothing can access the variable and it becomes more restricted than “private”

Correct!

### Question 40

1 / 1 pts

If you add a key/value combination to a Map and that key already exists in the Map, what happens?

☐ It will not compile

☐ An exception will happen because Map's are immutable

☒ The value in the map will be overwritten with the new value being passed

☐ The value in the map will be converted to a List and the one key will essentially have multiple values

Correct!

### Question 41

1 / 1 pts

Say you have an interface with one method and a different class makes it into a lambda. What happens if you add another method to the interface?

☐ Nothing special happens and the lambda can be created without issue

☐ It will compile fine as long as you have the `@FunctionalInterface` annotation on the interface

☒ It will not compile

☐ You will simply need to change the format of the lambda to account for the second method

Correct!

### Question 42

1 / 1 pts

Under what circumstances would you use the `ExecutorService`?

☐ If you need to execute code in a different thread

- ☐ If you need a class to be thread safe
- ☐ If you need to kill a particular thread
- ☒ If you need to manage multiple threads with certain limits

Correct!

### Question 43

1 / 1 pts

What exactly is a checked exception?

- ☒ An exception that needs to be caught or rethrown in order to compile
- ☐ An exception which cannot be caught
- ☐ An exception that extends RuntimeException
- ☐ An exception that will immediately cause the program to exit

Correct!

### Question 44

1 / 1 pts

What exactly is a JButton?

- ☒ Swing component for allowing users to initiate an action with a mouse click
- ☐ A heavy-weight Swing component used as a top-level container for other components
- ☐ The parent class of all Swing components
- ☐ Swing component for displaying simple text

Correct!

### Question 45

1 / 1 pts

Which of the following is true regarding Swing objects and concurrency?

Correct!

- ☒ Many Swing objects are not thread safe
- ☐ An exception will always happen if you try to create a Swing object on the main thread
- ☐ Whether a Swing object is thread safe or not depends on which thread created it
- ☐ Swing objects can safely be interacted with from multiple threads

### Question 46

1 / 1 pts

Which of the following is proper syntax for declaring a method that throws a checked exception?

Correct!

- ☐ `public void doStuff() throws RuntimeException {}`
- ☐ `public void doStuff() {} throws Exception`
- ☒ `public void doStuff() throws Exception {}`
- ☐ `public void doStuff() {} throws RuntimeException`

### Question 47

1 / 1 pts

What exactly is an anonymous class?

Correct!

- ☒ A class with no name that is declared and instantiated at the same time
- ☐ It is a class that has been marked with the “private” access modifier
- ☐ It is a special type of inner class that is not enclosed by another class
- ☐ It is the same as a static inner class

### Question 48

1 / 1 pts

What exactly does `SwingUtilities.invokeLater` do?

- ☐ Starts a new thread every time it is called
- ☐ Runs your code on the EDT and hangs until it and other queued tasks are finished
- ☒ Runs the given code on the EDT at some future point
- ☐ Runs the given code on the current thread but at a future point

Correct!

### Question 49

1 / 1 pts

Which of the following code snippets will correctly start a new thread?

- ☐ `new Thread(System.out.println("here")).start();`
- ☒ `new Thread(() -> System.out.println("here")).start();`
- ☐ `new Thread(System.out.println("here"));`
- ☐ `new Thread(() -> System.out.println("here"));`

Correct!

### Question 50

1 / 1 pts

What is the biggest difference between a Set and an ArrayList?

- ☐ Set can have items added to it, ArrayList cannot
- ☒ ArrayList allows duplicates but Set does not
- ☐ ArrayList implements Collection but Set does not
- ☐ Set is immutable, ArrayList is not

Correct!

### Question 51

1 / 1 pts

Which of the following statements is true?

- ☐ Initializing a variable is when a variable is declared at the class level, declaration is inline in a method
- ☐ There is no difference
- ☒ Initializing a variable is when you assign it a value, declaring a variable is when you first mention a variable in your code
- ☐ Initializing a variable is when you pass a parameter to a method, declaring a variable is when you pass something via a constructor

Correct!

### Question 52

1 / 1 pts

What exactly is an unchecked exception?

- ☐ An exception that does not propagate up
- ☐ An exception indicating the program is in an unusable state
- ☒ An exception that does not need to be caught or rethrown
- ☐ An exception that is closeable in a try-with-resources

Correct!

### Question 53

1 / 1 pts

Which of the following is a true statement?

- ☒ Everything that implements `java.util.Collection` can be used in a for-each loop
- ☐ Everything you can do with a `java.util.Collection` you can do with an array
- ☐ `Map` implements `Collection`
- ☐ A “Collection” and a “List” refer to the same thing in Java

Correct!

### Question 54

1 / 1 pts

What does it mean for an object to be immutable in Java?

- ☐ An object that can easily be extended
- ☒ An object whose internal state remains constant after creation
- ☐ An object that can be serialized
- ☐ An object that is inherently not safe for multi-threading

Correct!

### Question 55

1 / 1 pts

Given the following generic class Test, which of the following is a correct instantiation of Test of type Long?

```
public class Test<T> {}
```

☐ Test<> a = new Test<Long>();

☐ Test<String> a = new Test<>();

☒ Test<Long> a = new Test<>();

☐ Test a = new Test(Long.class);

Correct!

### Question 56

1 / 1 pts

How does Swing make UI components under the hood?

☐ It simply runs a Chromium web browser for rendering

☐ Swing does not make UI components - that's what AWT is for

☐ It uses native OS components for everything it does

☒ All it needs is the ability from the OS to paint pixels in an application frame and it builds its components within that frame

Correct!

### Question 57

1 / 1 pts



Which of the following is correct syntax for declaring a generic class?

☐ <K> public class Bla {}

☐ public class <K> Bla {}

☐ public <K> class Bla {}

☒ public class Bla<K> {}

Correct!

### Question 58

1 / 1 pts

Which of the following is the most commonly used implementation of java.util.Map?

☐ LinkedHashMap

☐ TreeMap

☐ Hashtable

☒ HashMap

Correct!

### Question 59

1 / 1 pts

Under what circumstances would a finally block not execute?

☐ If an exception is thrown inside the try block

☐ It will always execute 100% of the time

☒ It would only not execute if an abnormal program exit occurred

Correct!

- ☐ If an exception is thrown inside the catch block

### Question 60

1 / 1 pts

What will be the result of creating an abstract method inside of a class that is not marked as abstract?

Correct!

- ☒ It will not compile
- ☐ It simply forces subclasses to override the abstract method
- ☐ It will compile but when called the method will not execute any code
- ☐ If the method is called it will throw a StackOverflowError

Quiz Score: **59** out of 60