

## What you will be expected to do:

- Read and use an API.
- Determine the type of any Java expression. (*Example: What is the type of 'a' + 1 ?*)
- Distinguish between the *true type* and the *current type* for an object. (*Example: A a = new B();* )
- Write a basic class with fields, constructors, getter and setter methods, and generic types.
- Properly extend a class, including a constructor, inherited and overridden methods, and generic types.
- Write methods using variables, operators, if-statements, and loops.
- Write a loop manipulating an array, String, or linked list.
- Write code that catches and handles an exception.
- Use generic types correctly, both declaring them and specifying them.
- Write an anonymous class.

## Types you need to know:

int, double, char, boolean, Object, String, StringBuilder, wrapper classes, Comparable, Iterable, Iterator.

You need to be able to use these types as well as extend/implement Object, Comparable, Comparator, Iterable, and Iterator *without access to the API*. For all other types used on the exam, you will be given enough of the API to answer the question.

## Important Hint

You need to use types correctly. (For example, don't put a primitive to the left of a dot, don't use `==` to compare object contents, don't assign **null** to a primitive variable.)

Questions about the Java programming language that you should be able to answer and do.

What is a **data type**?

What is a typecast? When is it done automatically and when must the typecast be explicit? What differences and similarities are there in typecasting objects and typecasting primitive values? How many differences can you come up with between primitive types and objects?

How exactly are **int**'s, **double**'s, **char**'s and **boolean**'s represented? Why does a programmer need to know this? In what way is arithmetic on a computer different from "real" arithmetic?

What is the difference between **=**, **==**, and **.equals**?

What are variables and what are the rules you must follow when adding a variable to your program? What is the scope of a variable (when it exists and can be accessed)? What is the difference between class and instance fields, local variables and method parameters? What is the method call stack? What is the heap? What is stored in each? How do they work?

What is method overloading? What is method overriding? What is the difference between static and non-static methods? How do you create a recursive method?

How do you define a class? What is a constructor, what form does it have, and what is the order things are executed inside a constructor? What does **new** do?

How do you extend a class? What is **this** and **super**? How about **this()** and **super()**? What is *polymorphism*, i.e. what is the difference between the *true type* and the *current type* of an object, and how does that determine what method or variable is accessed? Why do we make some things public and some things private? What about protected?

When forming an object-oriented hierarchy, what is the role of the "is-a" and "has-a" relations and what are the rules for forming a good hierarchy? Can you explain the

ways in which an abstract class differs from an interface? What are the rules we should follow to take full advantage of Java's object oriented properties?

What is the API, and what is JavaDoc?

How do you concatenate strings? How do you access characters from a string? What is the end of line character? When is it okay to use the `+` operator and when should you use **StringBuilder**?

Why does Java have wrapper classes? What is wrapping and unwrapping (autoboxing and unboxing), and how, exactly, does it work?

How do you write a **while** loop? How do you write a **for** loop? How about a **foreach** loop? What rules should you use to test loops?

What is an array? How do you access an element from an array? How do you declare an array variable? How do you initialize an array? How do you copy an array? How do you insert into an array? What is a multi-dimensional array? How do you declare and initialize multi-dimensional arrays?

What is the difference between binary search and linear search? In which situations does binary search work better and when does linear search work better?

What is a linked list? How do you insert an element into a linked list? How do you remove an element? Can you give situations where a linked list is better than an array? Can you give situations where an array is better than a linked list?

What are generic types? How do you create a class that contains a generic? How do you instantiate objects of that class? How do you extend a class that contains a

generic? How and when do you use wildcards? How do you restrict what the generic can be?

What are exceptions? How do you create an exception? How do you throw an exception? How do you catch an exception? What is the difference between checked and unchecked exceptions? What is **finally** and how does it work? What is **final**?

What is the purpose of the **main** method? How does it work? How do you write it?

What are nested classes and anonymous classes and how do you create them? Why did we have to make some variables **final** (or effectively final) when using them in an anonymous class?

What is a Java ***lambda expression***? What is a ***method reference***? When can we use them? What is Java reflection and what can we use it for?

What are different types of errors and the "lessons learned" for avoiding them? What are ethical questions we should ask about a project? What techniques can we use to identify the ethical risks in a project?