**Test Automation Bootcamp**

Java Questions

1. **Explain what immutable objects are and why Strings are considered immutable objects.**
   1. **An immutable object is an**object whose internal state remains constant after it has been entirely created**.**

**Which means, once an object is created, we cannot change its content. In Java, all the wrapper classes (like Integer, Boolean, Byte, Short) and String class is immutable.**

* 1. **In** [String](https://dzone.com/articles/java-immutable-objects) **Objects class, String is considered as immutable Object with it replace method, the original String does not change.**
  2. **Being immutable automatically makes the String thread safe since they will not be changed when accessed from multiple threads. Hence immutable objects, in general, can be shared across multiple threads running simultaneously.**

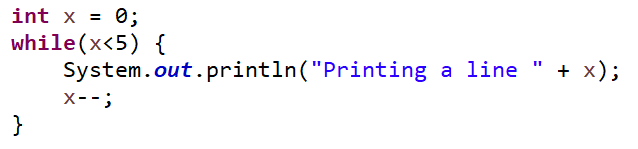
1. **Explain the problem that arises in Java with Multiple Inheritance.** 
   1. **Multiple inheritance is a feature where a class inherits a property from more than one parent class, which is not supported by Java.**

**The Problem arises when there exists the same method signature in both super classes and subclasses, hence the derived class will have to manage the dependency on two base classes.**

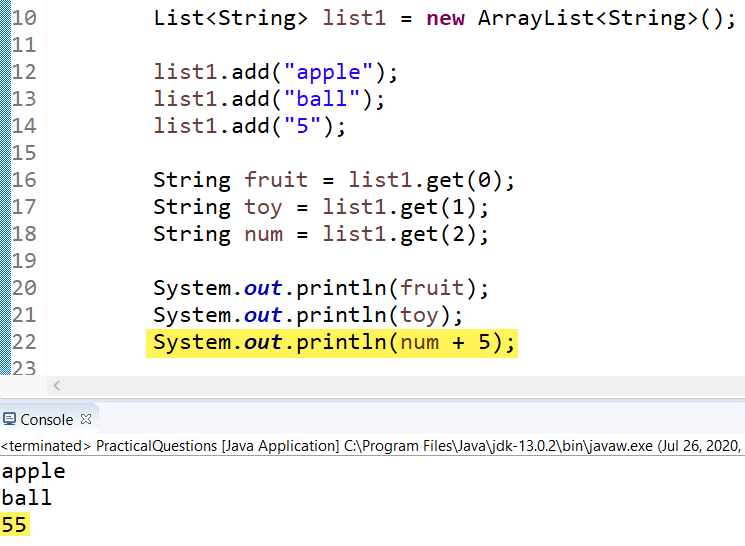
1. **What is the difference between an abstract class and an interface?**
   1. **Abstract class allows you to create functionality that subclasses can implement or override. A class can extends only one abstract class and implements multiple interfaces with implements keyword.**
   2. **An interface only allows you to define functionality, not implement it. And whereas a class can extend only one abstract class, it can take advantage of multiple interfaces.**
2. **Explain the difference between static & instance variables**
   1. **Static variable is a class variable created with the keyword static and can be used anywhere in the class without creating object of the class, created when the program runs and destroyed when the program stops.**

**Static variable a global variable shared among all objects within a class.**

* 1. **Instance variable is a class variable that can be accesses by creating an object of a class, they are created when the object is created with the keyword ‘new’ and destroyed when the object is destroyed.**

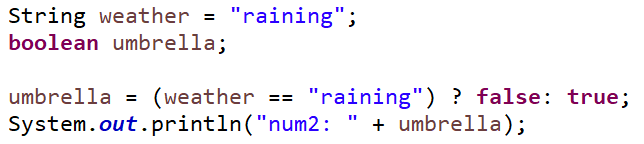
1. **What is the difference between the Throw & Throws keywords?**
   1. **Throw: is a keyword used to throw exception explicitly inside a block of code, could replace the try catch block.**
   2. **Throws: is a keyword used in a method signature to declare exceptions which might throw during runtime.**
2. **How can we implement a try block without implementing a catch block?**
   1. **Yes, It is possible to have a try block without a catch block by using a finally block,** a finally **block will** always **execute** even there is an exception occurred in a **try block.**
3. **What is the difference between a List and a Set?**
   1. List:
      1. Keeps the order of objects added to the list and the same when they are accessed.
      2. Duplicate values can be stored and retrieved, also allows null values.
   2. Set:
      1. Do not keep the order of objects added to the method.
      2. No duplicate value is allowed in the data set.
4. **Explain the problem with the code below:**
   1. **It is infinite loop that executes while the condition is true, hence the condition of the code below is always true because “x” is always “true” being less than 5.**
5. **How can you achieve a print-out that shows apple, ball, 10 in the console? See the code below:**
   1. **The console print** apple, ball, 10 **will be achieved by replacing the following line of code in the highlighted code.**

****

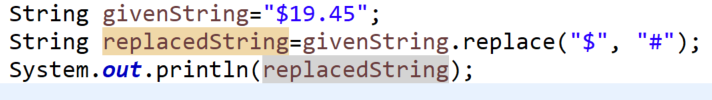


1. **What will be the output in the console of the code below?** 
   1. **The ternary operator code bellow will print in the console:**

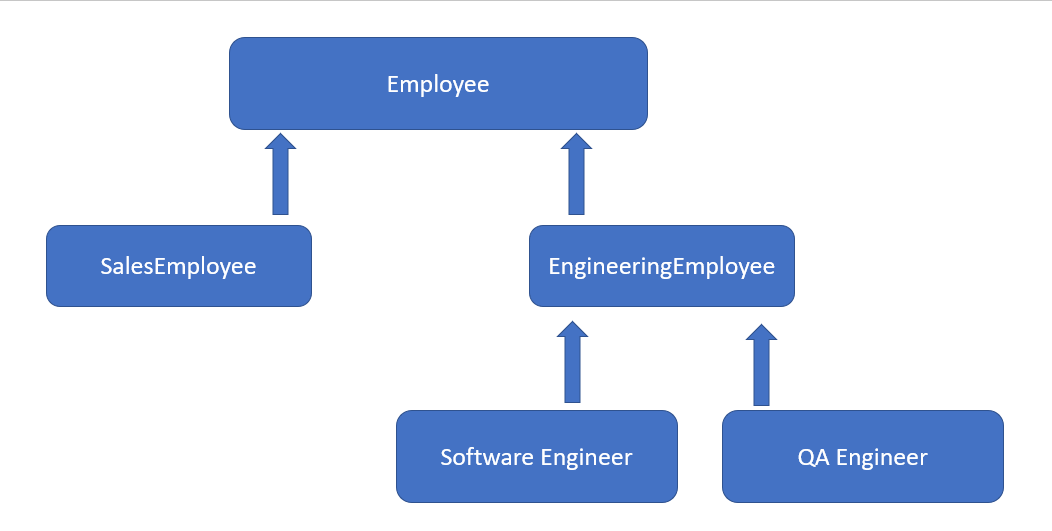
num2: false



1. **Explain the purpose of the break keyword.**
   1. **Break keyword will stop the iteration in a loop, and the program goes to the next line of code following the loop.**
   2. **It terminates a case in a switch statement.**
2. **Write the java code that will change the dollar sign ($) into a hash (#) in the following string: “$19.45”**
   1. **This code can replace the ($) to (#) as shown below:**



1. **What is the difference between == & the .equals String method?** 
   1. **== sign: is an equality operator used to compare primitive and Objects that checks for both objects point to the same memory location.**
   2. **.equals: is String method that compares string values in the Object.**
2. **Scenario: All employees are employees of a particular department – i.e. 1 employee is an employee of the Engineering department, another employee is an employee of the Sales department, etc. All engineering employees are either a Software Engineer or a QA Engineer. What type(s) of inheritance is described in this scenario?** 
   1. **Hybrid Inheritance: it is the result of single inheritance and Hierarchical inheritance.**



1. **What are getters & setters and what role do they play in Object Oriented Programming?** 
   1. **When a class variable is declared as Private, we do need to create getters and setters that allows access to the private method through a public accessor called getters and setters.**
   2. **They do an important role in securing the access of the private variable to be managed by the class only.** 
      1. Getters and setters are used to protect your data, particularly when creating classes. For each instance variable, a getter method returns its value while a setter method sets or updates its value.
      2. The getter method returns the value of the attribute.
      3. The setter method takes a parameter and assigns it to the attribute.
2. **Can we overload a method by changing the generic type of the method arguments?** 
   1. **No, The Generic type of the argument cannot be used as an overloading mechanism. But,**
   2. **A generic method may be overloaded like any other method. A class can provide two or more generic methods that specify the same method name but different method parameters.**
3. **What is Early Binding?**
   1. **Is a static binding happened during the compile time.**
4. **Regarding Dynamic Polymorphism, what is the basis for determining which method should be called – the parent’s method or the child’s method?**
   1. It is the runtime binding and when a parent class references is used to refer a child class object, the type of the reference variable determines the methods that can be invoked.
   2. **The instance of the object we are creating in the child’s method determines which method signature should be called and decided at runtime.**
   3. It is the type of object that determines which version of the method would be called (not the type of reference) in this case it the Child’s Method.
5. **How do you override a static method?** 
   1. **A static method cannot be override, a static method in java is associated with the class.**
6. **Why can private methods not be marked as abstract methods?**
   1. **Private methods are accessible with in a class only, so we cannot mark them as abstract since abstract methods are accessible through inheritance. It would be impossible for subclasses to implement a private abstract method.**
   2. The abstract method in the class can only set a visibility modifier one of public or protected.
7. **What are Constructors?**
   1. **Constructors are a special type of methods within a java class having the same name as class name, used to initialize a newly created object of that type.**
   2. **Can be overloaded with different method signature.**
   3. **Constructors are called whenever an instance of the class is created.**
8. **What is the difference between the *this* keyword and the *super* keyword?**
   1. **“this” keyword: is used to access methods and static members of a current class objects.**
   2. **“super” keyword** is used to access the methods or data members of the parent class.
9. **Is there a problem with the code shown below? If yes, what is the problem and what exception will be thrown?**
   1. **The main method could be written as “public static** **void** main(String[] args)”{ But, Still the main method compiles without exception.
   2. **NOTE: When compiling we should include the body closing brass, “}”**.
10. **What class is the parent of all other classes in Java?** 
    1. **Object Class: all other classes are a child of object Class.**
11. **What are some of the interfaces within the Collections Framework?**
    1. **The Collections framework interfaces that are present in java.util package are:**
       1. **List Interface**
       2. **Iterator and ListIterator**
       3. **ArrayList Interface**
       4. **HashMap Interface**
       5. **HashSet Interface**
       6. **Map interface**
       7. **Comparable and Comparator Interface**
       8. **LinkedList Interface**
       9. **TreeMap Interface**
       10. **HashTable Interface**
       11. **Vector Interface**