Technical Interviews 101

Module 0: Language Specifics (`https://jishanshaikh4.github.io/tech101/0`)

Compiled by: Jishan Shaikh. Last Updated December 11, 2019

What is expected?

Every candidate is expected to have some prior knowledge of at least one programming language; preferably an object-oriented one. Most popular languages among candidates are C/C++, Python, and Java. You can pick whichever language you want; You can expect most questions from that language only. However I recommend familiarity with multiple languages.

Usual Language Specifics' questions are of 2 types:

- **Language Constructs**
  - Common Syntax and Concepts
  - Code Reviews/Debuggings/Outputs
  - Code Implementation/Design
- **Inter-language questions:** Comparisons, barriers, feature-bug, and knowledge-based questions.

Some interviewers don't ask direct language questions, but some do. Preparation of these will also serve as prerequisite to other sections indirectly. Exemplers:

1. Implement a singleton class in C++/Python/Java?
2. Compiled vs Interpreted? Static- vs Dynamic-typing? Preference: Java or Python or C++? Why?
3. **Funny:** Is C++/Java/PHP going to dead soon?

Object Oriented Programming

What is it? Programming paradigm in which real world entitles are modeled as objects.

Class: A class is the blue-print of a real life entity.

Object: An object is an instance of a class.

Data Members: Data that can be stored in a class.

Methods/Functions: Explicitly written manipulators that can process data members.

Accessibility: Visibility/Usage of data members (or methods) from base/child/grand-child classes.

Inheritance: Taking properties from predefined structures. Types: Single, Multiple, Multilevel, Hierarchical, and Hybrid. ;

Abstraction: Showing only relevant information i.e. interfaces, and hiding irrelevant implementation details. Supported features for abstraction e.g. interface in Java, virtual functions in C++, interfaces/packages in Python.

Polymorphism: Multiple usage of single type name. Types: Compile-time (Operator overloading) and Runtime (Method overriding).

Encapsulation: The process of wrapping or putting up of data in to unit class and keeps data safe from misuse.

Bonus: Is C++/Python/Java pure object oriented?

Fun: What is a class diagram in C++/Python/Java?

C++ Programming

C++ concepts include constructors and destructors, copy constructors, virtual functions, Inheritance, Polymorphism, Classes and Objects, Memory Management (`new` and `delete`), Exception handling, Pointers and References, Strings, Operator Overloading, C vs C++, friend functions, Standard STL elements (vector, queue, list, etc.), Keywords, etc.

Sample Questions (Conceptual and Syntactical)

1. Reference in C++? Difference with a pointer?
2. Can we have virtual constructors and destructors?
3. What is the `sizeof` result for an empty class having no data members and no member functions? What if it has only constructors and destructors?
4. Have you heard of Named constructor Idiom?
5. Problems with runtime type identification.
6. What's the order that local objects are destructed?
7. What are copy constructors? When are they called?
8. The parameter in the copy constructor is always passed by pass-by-reference (**True**/False).
9. What are your views on Virtual Inheritance.
10. Implement a class `customString` having assignment and comparison operators.

Python Programming

Python provides heavy features and libraries with simple syntax in an interpreted language. Broad questions may be asked from Python section as it has surplus applications in scientific computing, web development, and machine learning. Next section contains some open-ended questions for Python Programming.

Sample Questions (Open-ended)

1. Arguments: Pass-by-reference or Pass-by-value?
2. Which Python3 libraries have you used previously?
3. ”A certain Lambda expression forms a closure”?
4. What will be the output of the following code?
   ```
   list = [’a’, ’b’, ’c’, ’d’, ’e’]
   print list [10:]
   ```
5. Describe the exceptional behaviour of `else` block after for & while loops. Is it a feature or a bug?
6. Use `super` or call `__init__` of parent class?
7. Does 3.000000000000004 problem exists in Python?
8. Is `rand()` in Python actually random?
9. Write a REGEX to detect an SQL injection attack!

Java Programming

Make sure you are familiar with all advanced constructs such as Java threads, collections, garbage collectors, exception handlings, Applets, Swing (GUI development), Servlets, RMI (Remote Method Invocation), JSP (Java Server Pages), etc. Android Developement experience is also a plus point for Java programming.

Sample Questions (General)

1. Differntiate: runtime and checked exceptions.
2. How do you ensure size of a primitive data type?
3. Difference between an interface & an abstract class.
4. How do you ensure that N threads can access M resources without deadlock? [Hint: Cases for greater M and N]
5. Is this possible in Java? `A extends B, C`
6. Differentiate: Enumeration & Iterator interfaces?
7. Anonymous inner class vs singleton class.
8. What are the alternatives of Java Serialization?
9. Java Sockets? Implementation and Advantages.
10. Compare the use-cases of `Equals()` & `==` in Java.

Recommended Resources

[1] *C++ Pocket Reference.* Kyle Louden. O'Reilly.

[2] *Programming Python.* Mark Lutz. O'Reilly.

[3] *Java Cookbook.* Ian F. Darwin. O'Reilly.

[4] *Programming Pearls.* John Bentley.

[5] *How to think like a computer scientist.* Mao Joe.

[6] *Effective Python.* Brett Slatkin. AW Publishing.

[7] *C++ Coding Standards.* Sutter and Alexandrescu.

Note: Go through a formal course in your preferable language. Also check the `Resources` folder of this module.

Know More

- Asking right on-the-spot questions is an art.
- Interviewing is a 2-way process; also ask interviewer for relevant, clarifying in-between questions.
- Interviewers intentionally ask vague questions to let candidate ask. The ability to ask questions to eliminate misunderstanding is a demonstration of effective communication & thinking skills.
- Think out loud. Sometimes interviewer is interested in how you are approaching a difficult problem than to get the right answer.
- Sometimes traditional solutions are rejected; Go for unusual perspectives and creative ideas in that case.