# C++ Beginner-Friendly Slide Deck

## Slide 1 — Welcome to C++ 🚀

* Powerful, general-purpose programming language
* Used in: systems, games, finance, embedded, AI
* Focus: learn step-by-step, practice, build projects

**Clickable resource:** [LearnCpp.com](https://www.learncpp.com)

## Slide 2 — Your Learning Setup ⚙️

* Install compiler: g++ / clang++
* Editor: VS Code (free) + C++ extension
* Online options: [Replit](https://replit.com), [OnlineGDB](https://www.onlinegdb.com)

**Docs:** [cppreference](https://en.cppreference.com)

## Slide 3 — Beginner Roadmap 🗺️

1. Basics: variables, types, operators, I/O
2. Control flow: if/else, loops, switch
3. Functions
4. Arrays & strings
5. Pointers & references

**Practice:** [Learn-Cpp.org](https://www.learn-cpp.org)

## Slide 4 — Moving Forward ➡️

* Classes & OOP (objects, constructors, destructors)
* STL containers (vector, map, string)
* Templates & generics
* Smart pointers & RAII
* Modern features: auto, range-for, lambdas

**Resource:** [C++ Core Guidelines](https://isocpp.github.io/CppCoreGuidelines/CppCoreGuidelines)

## Slide 5 — Projects for Beginners 💡

* Calculator
* Number guessing game
* To-do list (save to file)
* Contact manager

**Practice:** [HackerRank C++](https://www.hackerrank.com/domains/tutorials/10-days-of-cpp)

## Slide 6 — Debugging & Tools 🔧

* gdb (debugger)
* Sanitizers (find bugs)
* Valgrind (memory checker)
* CMake (build system)

**Docs:** [GCC manual](https://gcc.gnu.org/onlinedocs/)

## Slide 7 — Learning Resources 📚

* [cppreference](https://en.cppreference.com)
* [LearnCpp.com](https://www.learncpp.com)
* [The Cherno YouTube](https://www.youtube.com/c/TheCherno)
* [freeCodeCamp C++ course](https://www.youtube.com/watch?v=vLnPwxZdW4Y)
* [GeeksforGeeks C++](https://www.geeksforgeeks.org/c-plus-plus/)

## Slide 8 — Practice & Growth 🏆

* Problem solving: [LeetCode](https://leetcode.com), [Codeforces](https://codeforces.com)
* Contribute: explore C++ projects on [GitHub](https://github.com)
* Stay updated: [isocpp.org](https://isocpp.org)

## Slide 9 — Tips for Success ✨

* Code daily — even small problems
* Write projects, not just exercises
* Use modern C++ features
* Debugging is learning
* Join communities (Reddit r/cpp, Stack Overflow)

## Slide 10 — Next Steps 🚀

✅ Start with basics at [LearnCpp.com](https://www.learncpp.com)  
✅ Try projects (calculator, todo list)  
✅ Practice problems on HackerRank  
✅ Explore open-source on GitHub

**Goal:** Be comfortable writing and debugging small C++ projects in 4–6 weeks!

*(Export this as PDF for clickable links.)*