🚀 Mastering C++ Roadmap + Revision + Interview Prep

🟢 Foundations

Features of C++

- Combines procedural and OOP.
- Portability, performance, rich libraries.
- Low-level (pointers) + high-level (STL).

Concept Qs

- What are the main features of C++?
- Difference between compiler and interpreter?
- Difference between #define and const?

Practical Coding Qs

```
#include <iostream>

using namespace std;

int main() {

    cout << sizeof(int) << " " << sizeof(char) << " " << sizeof(double);

}
```

Guess the Output

```
int x = 10, y = 20;

cout << x + y << " " << x - y;
```

🟢 Program Structure + Compilation

- Header files → main() → body → return.
- Compilation steps: Preprocessing → Compilation → Linking → Execution.

Concept Qs

- Explain role of preprocessor.
- What happens in linking phase?

🟢 Core Language Skills

Constants, I/O, Operators

Concept Qs

- Difference between endl and \n.
- Difference between ++i and i++.

Practical Coding Qs

```
#include <iostream>

#include <iomanip>

using namespace std;

int main() {

   float pi = 3.14159;

   cout << fixed << setprecision(2) << pi;

}
```

Guess the Output

```
int a = 5;

cout << a++ << " " << ++a;
```

🟢 Flow Control

Decision Making + Loops

Concept Qs

- Difference between switch and if-else.
- Can switch work with string? (C++11 yes).

Practical Coding Qs

```
int n; cin >> n;

for(int i=1; i<=10; i++) cout << n*i << " ";
```

Guess the Output

```
for(int i=0; i<5; i++) {
   if(i==2) continue;
   cout << i;
}
```

🟢 Strings

Concept Qs

- Difference between C-style (char[]) and std::string.
- What does strcmp return if equal? (0).

Practical Coding Qs

```
string s = "hello";

reverse(s.begin(), s.end());

cout << s;
```

Guess the Output

```
char str[] = "abc";

cout << strlen(str);
```

🟢 Functions

Concept Qs

- Difference between call by value and reference.
- What are inline functions?

Practical Coding Qs

```
int fact(int n) {
   if(n==0) return 1;
   return n * fact(n-1);
}
```

Guess the Output

```
int f(int a, int b=5) { return a+b; }
cout << f(3) << " " << f(3,2);
```

🟢 Arrays

Concept Qs

- How are 2D arrays stored in memory?
- Time complexity of binary search?

Practical Coding Qs

```
int arr[] = {3, 9, 2, 8};
int max = arr[0];
for(int i=1;i<4;i++) if(arr[i] > max) max = arr[i];
```

Guess the Output

```
int arr[5] = {1,2,3};
cout << arr[3];
```

🟢 Pointers

Concept Qs

- Difference between NULL and nullptr.
- What is a dangling pointer?

Practical Coding Qs

```
void swap(int *a, int *b) {
   int t=*a; *a=*b; *b=t;
}
```

Guess the Output

```
int x=5; int *p=&x; *p=10;
cout << x;
```

🟢 User-Defined Types

Concept Qs

- Difference between struct in C and C++.
- Can struct have member functions? (Yes).

Practical Coding Qs

```
struct Student { int id; string name; };
```

Guess the Output

```
enum Color {Red, Green=5, Blue};
cout << Red << " " << Blue;
```

🟢 File Handling

Concept Qs

- Difference between ios::app and ios::out.
- How to check if file opened?

Practical Coding Qs

```
ofstream fout("data.txt");
fout << "Hello";
```

fout.close();

Guess the Output

ifstream fin("data.txt");

string s; fin >> s;

cout << s;

🟢 Error Handling

Concept Qs

- Syntax vs runtime vs logical errors.
- Why use try-catch?

🏛️ OOP Revision

OOP Basics

- Procedural vs OOP.
- Principles: Encapsulation, Abstraction, Inheritance, Polymorphism.

Concept Qs

- What is encapsulation?
- Real-world example of inheritance?

Classes & Objects

Practical Coding Qs

```
class Student {
   string name;
public:
   Student(string n) { name=n; }
   void display() { cout << name; }
};
```

Guess the Output

class Base { public: virtual void show(){cout<<"Base";} };

class Derived: public Base { public: void show(){cout<<"Derived";} };

Base *b = new Derived();

b->show();

👉 Output: Derived

Constructors & Destructors

Concept Qs

- Difference between constructor and normal function.
- What if destructor is missing?

Guess the Output

class Test {

public:

  Test(){ cout<<"Ctor "; }

  ~Test(){ cout<<"Dtor "; }

};

int main(){ Test t; }

Inheritance

Concept Qs

- Difference between function hiding and overriding.
- What is object slicing?

Practical Coding Qs

class A{ public: void show(){cout<<"A";} };

class B: public A{ public: void show(){cout<<"B";} };

Polymorphism

Concept Qs

- Compile-time vs runtime polymorphism.
- What is virtual destructor?

Guess the Output

class A { public: virtual void f(){cout<<"A";} };

class B: public A { public: void f(){cout<<"B";} };

A *a = new B(); a->f();

🟢 STL

Concept Qs

- Vector vs array.
- Does map allow duplicate keys? (No).

Practical Coding Qs

vector<int> v={1,2,3};

v.push_back(4);

for(auto x:v) cout<<x<<" ";

Guess the Output

set<int> s = {4,1,3,4};

for(int x:s) cout<<x<<" ";

🟢 Templates

Concept Qs

- Why templates are useful?
- Difference between overloading and template?

Practical Coding Qs

template<typename T>

T add(T a, T b) { return a+b; }

Guess the Output

cout << add(2,3) << " " << add(2.5,3.5);

⚡ How to Revise Daily

- Pick 1 topic.
- Revise notes (15 min).
- Solve 2 coding problems (30 min).
- Solve 2 guess-output problems (15 min).
- Move to next topic.