OOP with C++ Important Questions PDF

The most important OOP with C++ questions are procedural versus object-oriented programming, classes and objects, access specifiers, constructors and destructors including the copy constructor, static members, friend functions, inline functions, function and operator overloading, types of inheritance and the diamond problem, virtual base classes, virtual and pure virtual functions, abstract classes, templates, exception handling, file streams and the STL.

Key takeaways

- Operator overloading, virtual functions and inheritance programs carry the most marks.
- Learn the difference between compile-time and runtime polymorphism with one example each.
- Write complete, compilable programs with headers and a main function.
- These are practice questions, not past papers; check your own syllabus and official previous papers.

About these important questions

This page lists the most important Object Oriented Programming with C++ questions for B.Tech CSE and IT exams, arranged unit by unit. The units follow the standard syllabus that most Indian technical universities share, so the order may differ slightly from yours.

Please note: these are practice questions based on common syllabus topics. They are not verified past papers, and we do not claim that any question will appear in your exam. Check your own university syllabus first, then solve the official previous year papers. AKTU students can download them from the AKTU old question paper archive.

Unit 1: Introduction to OOP and C++ Basics

Covers OOP concepts, C++ basics, references and functions.

Short questions (2 marks)

- Differentiate between procedural and object-oriented programming.
- What is a reference variable?
- What is the scope resolution operator?
- What is an inline function?

Long questions

1. Explain the basic concepts of OOP with examples.
2. Explain the structure of a C++ program, and cin and cout.

3. Compare call by value, call by address and call by reference with programs.
4. Explain default arguments and inline functions with programs.
5. Explain dynamic memory allocation with new and delete.

Unit 2: Classes, Objects, Constructors and Destructors

Covers class definition, access specifiers, constructors, destructors, static members and friend functions.

Short questions (2 marks)

- Differentiate between a structure and a class in C++.
- What is a copy constructor?
- What is a friend function?

Long questions

1. Explain the types of constructors (default, parameterised, copy) with a program.
2. Explain destructors and the order in which constructors and destructors are called.
3. Explain static data members and static member functions with a program that counts objects.
4. Explain friend functions and friend classes with a program.
5. Explain the this pointer with an example.
6. Explain arrays of objects and passing objects to functions.

Unit 3: Polymorphism and Overloading

Covers function overloading, operator overloading and type conversion.

Short questions (2 marks)

- Which operators cannot be overloaded in C++?
- Differentiate between compile-time and runtime polymorphism.

Long questions

1. Explain function overloading with a program.
2. Overload the binary + operator to add two complex numbers, using a member function and using a friend function.
3. Overload the unary minus or the ++ operator (prefix and postfix).
4. Overload the << and >> operators for a class.
5. Explain type conversion: basic to class, class to basic and class to class.

Unit 4: Inheritance and Virtual Functions

Covers inheritance types, ambiguity, virtual base classes, virtual functions and abstract classes. This unit is asked heavily.

Short questions (2 marks)

- List the types of inheritance.

- What is a pure virtual function?
- What is an abstract class?

Long questions

1. Explain single, multiple, multilevel, hierarchical and hybrid inheritance with programs.
2. Explain the diamond problem and how a virtual base class solves it.
3. Explain the visibility of members in public, private and protected inheritance with a table.
4. Explain virtual functions and runtime polymorphism with a base class pointer.
5. Explain pure virtual functions and abstract classes with a Shape example.
6. Explain virtual destructors and why they are needed.

Unit 5: Templates, Exceptions, Files and STL

Covers generic programming, exception handling, file streams and the Standard Template Library.

Short questions (2 marks)

- What is a template?
- List the three keywords used in C++ exception handling.
- What is a container in STL?

Long questions

1. Explain function templates and class templates with programs.
2. Explain exception handling with try, throw and catch, including multiple catch blocks.
3. Write a program to write to and read from a file using ofstream and ifstream.
4. Explain file modes and random access with seekg and tellg.
5. Explain the components of the STL: containers, iterators and algorithms, with a vector example.
6. Compare vector, list and map.

Sample answer outline: Overloading the + operator for complex numbers

This is one of the most common C++ long questions. A full answer has these parts:

1. **Define operator overloading.** It lets an operator such as + work with user-defined types, giving it a new meaning for that class. It is a form of compile-time polymorphism.
2. **State the syntax.** `return_type operator+(argument)` as a member function, or as a friend function with two arguments.
3. **Write the program.** A class Complex with private real and imag members, a constructor, an `operator+` member function that returns a new Complex with the sums, and a display function. In main, create c1 and c2, compute `c3 = c1 + c2` and display c3.
4. **Explain the call.** `c1 + c2` is translated to `c1.operator+(c2)`, so c1 is the calling object and c2 is the argument.
5. **Add the friend version.** With `friend Complex operator+(Complex, Complex)`, both objects are passed as arguments. This form is needed when the left operand is not an object of the class.
6. **List rules.** You cannot overload ::, ., .*, ?: or sizeof, and overloading cannot change precedence.

Marks weightage and how to write answers

A common pattern, used by AKTU and many other universities, has a compulsory short-answer section that covers every unit, followed by long-answer questions with a choice inside each unit. Marks per question change between universities and years, so confirm the scheme on a recent official paper.

- **Short answers (about 2 marks):** a crisp definition, one example or diagram, and nothing more. Three to five lines is enough.
- **Long answers (7 to 10 marks):** start with a definition, add a labelled diagram or table, explain the steps, and finish with an example, advantages or a comparison.
- Include headers, using namespace std (or std::), a class and a main function in every program.
- For inheritance questions, draw the class hierarchy diagram before the code.

A 7-day revision plan for OOP with C++

1. **Days 1 to 5:** one unit a day. Read your notes, then write the long answers for that unit from memory and check them against your notes.
2. **Day 6:** write from memory the programs for a copy constructor, + operator overloading, a virtual base class, virtual functions and a class template.
3. **Day 7:** solve one full official previous year paper in three hours, then check which units cost you time.

Keep a single sheet of formulas, definitions and diagrams for the last hour before the exam.

Frequently asked questions

Which OOP with C++ topics are most important?

Constructors and destructors, friend functions, static members, operator overloading, inheritance and the diamond problem, virtual functions, abstract classes, templates and exception handling are asked most often.

What is the difference between C++ and Java OOP?

C++ supports multiple inheritance with classes, operator overloading, pointers and manual memory management. Java uses interfaces for multiple inheritance, has no operator overloading and manages memory with garbage collection.

Is C++ useful for placements?

Yes. Many students use C++ for coding rounds because of the STL, and OOP questions are common in interviews.

Are these OOP with C++ questions from previous year papers?

No. They are practice questions built from topics that appear in most B.Tech Object Oriented Programming with C++ syllabi. Use them with your own university's official previous papers, for example the AKTU archive at https://aktu.ac.in/old-question-paper.html.

Is this PDF free?

Yes. Fill in the short form on this page and the PDF downloads right away.

Sources and further reading

- AKTU: old question papers (official archive)
- cppreference: C++ language reference
- NPTEL: online courses from IITs (search for Programming in C++)

About this PDF

These are practice questions for revision, prepared by the Uncodemy trainer team from common B.Tech syllabus topics. They are not past papers, a certificate or a promise of marks. Spot an error? Write to info@uncodemy.com and the team will review it.

More free resources and courses: uncodemy.com | info@uncodemy.com | +91 98183 66550