1. Explain the need for Object-Oriented Programming (OOP). How does it differ from procedural programming?
2. What is Object-Oriented Programming (OOP)? Explain its advantages over procedural programming.
3. List and explain the four main pillars of OOP: Encapsulation, Inheritance, Polymorphism, and Abstraction.
4. What are objects and classes in OOP? How are they related?
5. Write a short note on a message passing in OOP.
6. Why is reusability an important feature of OOP?
7. List and explain the key features of OOP with suitable examples.
8. Discuss data abstraction and encapsulation with real-life examples.
9. Explain the concept of inheritance in OOP. How does it promote reusability?
10. Explain the structure of a C++ program with an example
11. What are built-in and user-defined data types in C++? Explain with examples.
12. Discuss access specifiers in C++ (private, public, protected) with examples.
13. How is C++ different from C? Explain the key differences.
14. Write a C++ program to create a class and object demonstrating encapsulation.
15. What is a constructor? Explain its purpose with an example.
16. What is a default constructor? How does it work? Give an example.
17. Explain parameterized constructors with an example.
18. What is a copy constructor? How does it differ from assignment operator?
19. Explain the use of default arguments in C++ with a constructor example.
20. What is a destructor? Why is it important in C++? Provide an example.
21. How does a constructor differ from a destructor? Discuss with an example.
22. Discuss the advantages and disadvantages of using constructors and destructors in C++.
23. Write a program to demonstrate the use of a copy constructor in C++.
24. Write a C++ program to demonstrate class and object creation.
25. Implement a C++ program to demonstrate encapsulation using private data members and public member functions.
26. Write a program to illustrate access specifiers (private, protected, and public) with proper examples.
27. Create a C++ program that defines a class "Student" with attributes (name, roll number, marks) and member functions to accept and display details.
28. Write a program to differentiate between procedural and object-oriented programming using C and C++ examples.
29. Implement a default constructor in a class and show how it is called automatically.
30. Write a C++ program to implement parameterized constructors and initialize object values using user input.
31. Implement a copy constructor to copy data from one object to another and display the copied values.
32. Write a program to demonstrate the use of destructors and observe when they are called.

33. Create a class "Rectangle" with a parameterized constructor, and find the area by initializing values through a constructor.
34. Implement a C++ program to copy the contents of one object to another using a copy constructor.