C++ 100 interview questions and answers.

1. What is C++?

Answer: C++ is a general-purpose, object-oriented programming language used for developing software that can be used across multiple platforms and devices. C++ is an extension of the C programming language and was created by Bjarne Stroustrup in 1983. C++ is a powerful language that can be used to create efficient and powerful applications.

2. What is the benefit of using C++?

Answer: C++ has many benefits, including its ability to create high-performance applications, its flexibility, its cross-platform capabilities, its object-oriented programming features, its large standard library, and its robust support for the development of graphical user interfaces. C++ is also a highly portable language, meaning that it can be used to create applications for a variety of different platforms and devices.

3. What is the difference between C and C++?

Answer: C is a procedural programming language, while C++ is an object-oriented programming language. C++ is also much more powerful than C, offering features such as classes, operators, exception handling, and templates. C++ also supports the development of graphical user interfaces and provides a much larger standard library than C.

4. What is the difference between a class and an object?

Answer: A class is a template that describes the characteristics of an object. An object is an instance of a class, meaning that it is a concrete representation of a particular class. Objects can have their own data and methods that are specific to that particular object.

5. What is an abstract class?

Answer: An abstract class is a class that cannot be instantiated, meaning that it cannot be used to create an object. An abstract class can have abstract methods, which are methods that have no implementation and must be implemented by a subclass. Abstract classes are used to define an abstraction, or a common interface, for a group of related classes.

6. How is data encapsulation implemented in C++?

Answer: Data encapsulation is implemented in C++ through the use of classes. A class can be used to define the data and functions that make up the class, and all of this data and functions are then encapsulated within the class. This means that the data and functions of the class can only be accessed through the class, thus preventing outside code from directly accessing or modifying the class's data.

7. What is the difference between an interface and an abstract class?

Answer: An interface is a collection of method signatures that can be implemented by a class. An abstract class is a class that cannot be instantiated and provides an implementation of some or all of the methods defined in an interface. An abstract class can also contain additional data and methods that are not present in the interface.

8. What is polymorphism?

Answer: Polymorphism is the ability of an object to take on different forms. In C++, polymorphism is implemented through the use of virtual functions, which allows a derived class to override a method from its base class. Polymorphism also allows for the dynamic binding of objects, meaning that an object can be bound to different classes at runtime.

9. What is the difference between a reference and a pointer?

Answer: A reference is an alias for another object, meaning that it is essentially a pointer that cannot be modified. A pointer is a variable that stores the address of another object and can be used to access and modify the data of that object. Both references and pointers can be used to refer to the same object, but references cannot be modified while pointers can.

10. What is the difference between a struct and a class?

Answer: A struct is a user-defined type that is used to group related data together into a single entity. Structs are similar to classes, but classes support inheritance and can contain methods, while structs cannot. Structs are usually used to represent simple data types, while classes are used to represent more complex entities.

11. What are the advantages of using C++?

Answer: The advantages of using C++ include its ability to create high-performance applications, its flexibility, its cross-platform capabilities, its object-oriented programming features, its large standard library, and its robust support for the development of graphical user interfaces. C++ is also a highly portable language, meaning that it can be used to create applications for a variety of different platforms and devices.

12. What is the difference between a constructor and a destructor?

Answer: A constructor is a special member function that is called when an object is created. A constructor can be used to initialize an object's data members and to allocate memory for the object. A destructor is a special member function that is called when an object is destroyed. A destructor can be used to deallocate memory or to clean up any resources that were allocated by the constructor.

13. What is the Standard Template Library (STL)?

Answer: The Standard Template Library (STL) is a collection of generic algorithms, containers, and iterators that can be used to create efficient and optimized C++ programs. The STL provides a wide range of functionality that can be used to create robust and efficient applications.

14. What is the difference between function overloading and function overriding?

Answer: Function overloading is the ability to have multiple functions with the same name, but with different parameters. Function overriding is the ability to redefine a virtual function in a derived class in order to provide a different implementation than the one provided by the base class.

15. What is a memory leak?

Answer: A memory leak is a condition in which memory that has been allocated for a program is not released, resulting in the program becoming slower and using more memory over time. Memory leaks can be caused by forgetting to deallocate memory that was allocated or by not properly handling errors that can cause the memory to not be released.

16. What is the difference between a deep copy and a shallow copy?

Answer: A deep copy is a copy of an object that contains all of the data from the original object, including any pointers and references that the original object had. A shallow copy is a copy of an object that only contains the data that is directly accessible by the original object. References and pointers are not copied in a shallow copy.

17. What is the difference between an array and a vector?

Answer: An array is a fixed-size collection of elements of the same type. A vector is a dynamic array that can grow and shrink as elements are added or removed. Vector also provides additional features such as the ability to access elements using iterators, as well as the ability to sort and search elements.

18. What is an iterator?

Answer: An iterator is an object that provides a way to access the elements of a container in a sequential order. Iterators can be used to traverse a container and access the elements of the container. Iterators also provide the ability to modify the elements of a container.

19. What are the benefits of using namespaces?

Answer: Namespaces provide a way to organize code and to avoid name collisions. Namespaces are used to group related code together, making it easier to find and use. Namespaces also provide a way to avoid name collisions between variables and functions that have the same name.

20. What is the difference between a static member and a non-static member?

Answer: A static member is a member of a class that is shared by all instances of the class. A non-static member is a member of a class that is unique to each instance of the class. Static members are typically used to store data that is shared by all instances of the class, while non-static members are typically used to store data that is unique to each instance of the class.

21. What is the difference between an inline function and a normal function?

Answer: An inline function is a function that is expanded in place when it is called, meaning that the code of the function is inserted directly into the code of the calling function. A normal function is a function that is invoked when it is called and is not expanded in place.

Inline functions are typically used to improve performance, while normal functions are used to modularize code.

22. What is the difference between a public and a private member?

Answer: A public member of a class can be accessed from outside of the class, while a private member can only be accessed from within the class. Public members are typically used to provide access to the data or methods of a class, while private members are typically used to hide data or methods from outside code.

23. What is the difference between an array and a linked list?

Answer: An array is a fixed-size collection of elements of the same type, while a linked list is a dynamic collection of elements of the same type. An array allows for random access of its elements, while a linked list does not. An array also requires that the elements of the array be stored in contiguous memory, while a linked list does not have this requirement.

24. What is the difference between a stack and a queue?

Answer: A stack is a data structure that provides Last In First Out (LIFO) access to its elements, while a queue is a data structure that provides First In First Out (FIFO) access to its elements. Stacks can be used to implement recursive algorithms, while queues can be used to implement asynchronous communication between threads.

25. What is a binary search tree?

Answer: A binary search tree is a type of data structure that allows for faster search operations than linear data structures such as linked lists, arrays, and stacks. Binary search trees are organized in a tree-like structure, with each node containing a key, value, and two children, the left and right nodes. The left node contains a key that is smaller than the parent node's key, and the right node contains a key that is larger than the parent node's key. To search for a value, the tree is traversed starting from the root node, and depending on the value of the key, either the left or right subtree is searched. This process is repeated until the desired value is found.

Coding Question:

Write a function to find the largest element in a binary search tree.

Solution:

```
// Function to find the largest element in a binary search tree
int findLargest(Node* root)
{
   // Base case
   if (root == NULL)
      return INT_MIN;

   // Return maximum of 3 values:
   // 1) Root's data 2) Max in Left Subtree
   // 3) Max in right subtree
   int res = root->data;
   int lres = findLargest(root->left);
   int rres = findLargest(root->right);
   if (lres > res)
      res = lres;
   if (rres > res)
      res = rres;
   return res;
}
```

26. What is a linked list?

Answer: A linked list is a data structure that consists of a sequence of nodes, each of which stores data and a pointer to the next node in the list. Linked lists are used to store and traverse data in a linear fashion.

27. What is a stack?

Answer: A stack is a data structure that works on a last-in-first-out (LIFO) basis. Elements can be added to the top of the stack and removed from the top of the stack, allowing for efficient operations such as push, pop, and peek.

28. What is a queue?

Answer: A queue is a data structure that works on a first-in-first-out (FIFO) basis. Elements are added to the end of the queue and removed from the beginning of the queue, allowing for efficient operations such as enqueue and dequeue.

29. What is a hash table?

Answer: A hash table is a data structure that stores data in key-value pairs. A hash function is used to generate a unique key for each item, which is then used to store the item in the table. Hash tables are used for fast lookups, since the time required to find an item is constant regardless of the number of items in the table.

30. Write a C++ program to implement a binary search tree.

Answer:

```
#include <iostream>

struct Node {
  int data;
  Node *left;
  Node *right;
};

// Function to create a new Node in the tree
Node *newNode(int data) {
  Node *node = new Node;
  node->data = data;
  node->left = NULL;
```

```
  node->right = NULL;
  return node;
}

// Function to insert a new Node in the tree
void insertNode(Node *&root, int data) {
  if(root == NULL) {
    root = newNode(data);
    return;
  } else if(data <= root->data) {
    insertNode(root->left, data);
  } else {
    insertNode(root->right, data);
  }
}

// Function to search for an element in the tree
bool search(Node *root, int data) {
  if(root == NULL) {
    return false;
  } else if(root->data == data) {
    return true;
  } else if(data <= root->data) {
    return search(root->left, data);
  } else {
    return search(root->right, data);
  }
```

```
}

int main() {
  // Create an empty tree
  Node *root = NULL;

  // Insert nodes into the tree
  insertNode(root, 15);
  insertNode(root, 10);
  insertNode(root, 20);
  insertNode(root, 25);
  insertNode(root, 8);
  insertNode(root, 12);

  // Search for a node with value 25
  if(search(root, 25)) {
    std::cout << "Found!" << std::endl;
  } else {
    std::cout << "Not found!" << std::endl;
  }

  return 0;
}
```

31. What is a heap?

Answer: A heap is a special kind of tree-based data structure that satisfies the heap property, which states that each node in the heap is either greater than or equal to its

parent node, or is a leaf node. Heaps are used for efficient operations such as insertion, deletion, and finding the maximum or minimum value in a collection.

32. What is the difference between an array and a linked list?

Answer: The main difference between an array and a linked list is that an array stores data contiguously in memory, while a linked list stores data non-contiguously in memory. Additionally, an array has a fixed size, whereas a linked list can grow and shrink in size as needed.

33. Write a function in C++ to delete a node in a linked list given the reference to the node to be deleted.

Answer:

```
// Function to delete a node in a linked list
void deleteNode(Node* node)
{
  // Store the next node
  Node* next = node->next;

  // Copy the data from the next node to the node to be deleted
  node->data = next->data;

  // Unlink the node from linked list
  node->next = next->next;
  // Delete the next node
  delete next;
}
```

34. What is a priority queue?

Answer: A priority queue is a type of data structure where each element has a priority associated with it. Elements are added and removed according to their priority, with the highest priority elements being added and removed first. Priority queues are used for efficient operations such as finding the maximum or minimum value in a collection.

35. What is a graph?

Answer: A graph is a data structure that consists of a set of nodes (vertices) connected by edges. Graphs can be used to represent relationships between objects, as well as to efficiently store and traverse data.

36. What is the difference between a depth-first search and a breadth-first search?

Answer: A depth-first search is a type of search algorithm that starts at the root node and explores as far as possible along each branch before backtracking. A breadth-first search is a type of search algorithm that starts at the root node and explores all of the neighboring nodes before moving on to the next level of nodes.

37. What is a dynamic programming algorithm?

Answer: A dynamic programming algorithm is a type of algorithm that is used to solve complex problems by breaking them down into smaller subproblems. The subproblems are then solved and the solutions are stored in a table, which is used to solve the original problem. Dynamic programming algorithms are used to solve optimization problems such as the shortest path problem and the knapsack problem.

38. What is a greedy algorithm?

Answer: A greedy algorithm is a type of algorithm that makes decisions in a step-by-step fashion, always choosing the best immediate option. Greedy algorithms are used to solve optimization problems such as the traveling salesman problem and the minimum spanning tree problem.

39. What is an algorithm?

Answer: An algorithm is a set of instructions that is used to solve a problem. Algorithms are used to solve a wide variety of problems, including sorting, searching, and optimization.

Algorithms can be written in any number of programming languages, such as C++, Java, and Python. .

40. Write a C++ program to implement a bubble sort algorithm.

Answer:

```
#include <iostream>

using namespace std;


void bubbleSort(int arr[], int n)

{

  int i, j;

  for (i = 0; i < n-1; i++)


    // Last i elements are already in place

    for (j = 0; j < n-i-1; j++)

      if (arr[j] > arr[j+1])

        swap(arr[j], arr[j+1]);

}


/* Function to print an array */

void printArray(int arr[], int size)

{

   int i;

   for (i=0; i < size; i++)

      cout << arr[i] << " ";

   cout << endl;

}
```

```
// Driver program
int main()
{
   int arr[] = {64, 34, 25, 12, 22, 11, 90};
   int n = sizeof(arr)/sizeof(arr[0]);
   bubbleSort(arr, n);
   cout<<"Sorted array: \n";
   printArray(arr, n);
   return 0;
}
```

41. What is the Big O notation?

Answer: Big O notation is a way of expressing the complexity of an algorithm in terms of time and space. It is used to measure the asymptotic performance of algorithms in terms of the number of operations it takes for the algorithm to execute.

42. What is the difference between a stack and a heap?

Answer: A stack is a type of data structure where elements are added and removed in a Last-In-First-Out (LIFO) order. A heap is a type of data structure where elements are added and removed in a First-In-First-Out (FIFO) order.

43. What is an algorithm?

Answer: An algorithm is a set of instructions that is used to solve a problem. Algorithms are used to solve a wide variety of problems, including sorting, searching, and optimization. Algorithms can be written in any number of programming languages, such as C++, Java, and Python.

44. What is the difference between a recursive and iterative algorithm?

Answer: A recursive algorithm is one that calls itself to solve a problem, while an iterative algorithm solves a problem by repeatedly performing a set of instructions.

45. What is the difference between a static member and a non-static member?

Answer: A static member is a member that is shared by all objects of the same type, while a non-static member is a member that belongs to a specific object.

46. What is an abstract class?

Answer: An abstract class is a class that cannot be instantiated, but instead must be inherited from. Abstract classes are used as a base from which other classes can inherit from, and they typically contain abstract methods that must be implemented in the derived classes.

47. What is the difference between an interface and an abstract class?

Answer: An interface is a class that contains only abstract methods, while an abstract class can contain both abstract and non-abstract methods.

48. What is polymorphism?

Answer: Polymorphism is the ability of an object to take on multiple forms. In C++, polymorphism is achieved through inheritance, virtual functions, and operator overloading.

49. What is the difference between a reference and a pointer?

Answer: A reference is a variable that holds the address of another variable, while a pointer is a variable that holds the address of another variable and can also be used to manipulate the data stored at that address.

50. What is the Standard Template Library (STL)?

Answer: The Standard Template Library (STL) is a collection of containers, algorithms, and iterators that provide a generic, type-safe, and efficient way to manipulate data in C++.

51. What is a memory leak?

Answer: A memory leak is a type of resource leak that occurs when a computer program incorrectly manages memory allocations in such a way that memory which is no longer needed is not released.

52. What is the difference between a deep copy and a shallow copy?

Answer: A deep copy duplicates all the data of the original object, while a shallow copy only duplicates the reference to the original object.

53. What is an iterator?

Answer: An iterator is an object that provides sequential access to elements of a container, such as a vector or list.

54. What are the benefits of using namespaces?

Answer: The benefits of using namespaces include avoiding name collisions with identifiers defined in other libraries, providing a concise way of organizing code, and making code easier to read.

55. What is the difference between a static member and a non-static member?

Answer: A static member of a class is shared among all instances of the class, while a non-static member belongs to a specific instance of the class.

56. What is the difference between an inline function and a normal function?

Answer: An inline function is a function that is expanded in-line by the compiler, while a normal function is called from the code, resulting in a function call overhead.

57. What is the difference between a public and a private member?

Answer: A public member is accessible to any code outside of the class, while a private member is only accessible within the class.

58. What is the difference between an array and a vector?

Answer: An array is a fixed-size collection of elements of the same type, while a vector is a dynamic array that can grow as needed.

59. What is a linked list?

Answer: A linked list is a data structure that consists of a sequence of nodes, each containing a reference to the next node in the sequence.

60. What is a stack?

Answer: A stack is a data structure that follows the Last In First Out (LIFO) principle, meaning that the last element added to the stack is the first element removed.

61. What is a queue?

Answer: A queue is a data structure that follows the First In First Out (FIFO) principle, meaning that the first element added to the queue is the first element removed.

62. What is a hash table?

Answer: A hash table is a data structure that maps keys to values using a hash function, allowing for fast lookups and insertion of elements.

63. Write a C++ program to implement a binary search tree.

Answer:

```
#include <iostream>

using namespace std;

struct Node

{

   int key;

   Node* left;

   Node* right;

};


Node* newNode(int item)

{
```

```
    Node* temp = new Node;
    temp->key = item;
    temp->left = temp->right = NULL;
    return temp;
}

Node* insert(Node* node, int key)
{
    if (node == NULL) return newNode(key);

    if (key < node->key)
        node->left  = insert(node->left, key);
    else if (key > node->key)
        node->right = insert(node->right, key);

    return node;
}

int main()
{
    Node* root = NULL;
    root = insert(root, 50);
    insert(root, 30);
    insert(root, 20);
    insert(root, 40);
    insert(root, 70);
    insert(root, 60);
```

```
    insert(root, 80);
    return 0;
}
```

64. What is a heap?

Answer: A heap is a specialized tree-based data structure that satisfies the heap property, meaning that the value of each node is greater than or equal to the value of its parent node.

65. What is the difference between a depth-first search and a breadth-first search?

Answer: A depth-first search explores nodes in the order in which they are stored in memory, while a breadth-first search explores nodes in the order in which they are added to the search tree.

66. What is a dynamic programming algorithm?

Answer: A dynamic programming algorithm is an algorithm that solves a complex problem by breaking it into subproblems and storing the results of the subproblems to avoid computing the same result multiple times.

67. What is a greedy algorithm?

Answer: A greedy algorithm is an algorithm that makes the best choice at each step in order to optimize its overall performance.

68. What is the Big O notation?

Answer: The Big O notation is a way of expressing the complexity of an algorithm in terms of time and space.

69. Write a function in C++ to delete a node in a linked list given the reference to the node to be deleted.

Answer:

```
void deleteNode(Node* node)
{
   if (node == NULL)
      return;
   // Copy the data from the next node to this node
   node->data = node->next->data;
   // Store the reference to the next node
   Node* next = node->next;
   // Remove the link of the next node
   node->next = node->next->next;
   // Free memory for the next node
   delete next;
}
```

70. What is a priority queue?

Answer: A priority queue is a data structure that stores elements according to their priority, with the highest priority elements being removed first.

71. What is a graph?

Answer: A graph is a data structure composed of nodes and edges, where nodes represent objects or entities and edges represent relationships between them.

72. What is the difference between a stack and a heap?

Answer: A stack is a linear data structure that follows a Last In First Out (LIFO) ordering, while a heap is a hierarchical data structure that follows a First In First Out (FIFO) ordering.

73. What is an algorithm?

Answer: An algorithm is a set of instructions for solving a problem or performing a task.

74. What is the difference between a recursive and iterative algorithm?

Answer: A recursive algorithm is one that calls itself to solve a problem, while an iterative algorithm is one that uses a looping structure to solve a problem.

75. What is the difference between a static member and a non-static member?

Answer: A static member is a member of a class that is shared by all instances of the class and is accessible without an instance of the class, while a non-static member is a member of a class that is specific to each instance of the class and must be accessed through an instance of the class.

76. What is an abstract class?

Answer: An abstract class is a class that cannot be instantiated and is used as a base class for other classes.

77. What is the difference between an interface and an abstract class?

Answer: An interface is a collection of abstract methods that must be implemented by a class that implements the interface, while an abstract class is a class that cannot be instantiated and is used as a base class for other classes.

78. What is polymorphism?

Answer: Polymorphism is the ability of an object to take on multiple forms, allowing it to be used in different ways.

79. What is the difference between a reference and a pointer?

Answer: A reference is an alias for a variable, while a pointer is a variable that holds the address of another variable.

80. What is the Standard Template Library (STL)?

Answer: The Standard Template Library (STL) is a collection of algorithms and data structures that are commonly used in C++ programs.

81. What is a memory leak?

Answer: A memory leak is when a program allocates memory but fails to deallocate it, resulting in wasted memory.

82. What is the difference between a deep copy and a shallow copy?

Answer: A deep copy is a copy of an object and all of its members, while a shallow copy is a copy of the object without copying its members.

83. What is the difference between an array and a vector?

Answer: An array is a fixed-sized collection of elements stored in contiguous memory, while a vector is a variable-sized collection of elements stored in contiguous memory.

84. What is an iterator?

Answer: An iterator is an object that points to elements of a container and allows access to them.

85. What are the benefits of using namespaces?

Answer: Namespaces allow programmers to group related code into logical units, reducing the chance of introducing conflicts between different pieces of code.

86. What is the difference between a static member and a non-static member?

Answer: A static member is a member of a class that is shared by all instances of the class and is accessible without an instance of the class, while a non-static member is a member of a class that is specific to each instance of the class and must be accessed through an instance of the class.

87. What is the difference between an inline function and a normal function?

Answer: An inline function is a function that is expanded in-place at compile time, while a normal function is a function that is called at run time.

88. What is the difference between a public and a private member?

Answer: A public member is accessible to all parts of the program, while a private member is accessible only within its own class.

89. What is the difference between an array and a linked list?

Answer: An array is a fixed-sized collection of elements stored in contiguous memory, while a linked list is a variable-sized collection of elements stored in non-contiguous memory.

90. What is a linked list?

Answer: A linked list is a data structure composed of nodes, where each node contains a reference to the next node in the list.

91. What is a stack?

Answer: A stack is a linear data structure that follows a Last In First Out (LIFO) ordering.

92. What is a queue?

Answer: A queue is a linear data structure that follows a First In First Out (FIFO) ordering.

93. What is a hash table?

Answer: A hash table is a data structure that maps keys to values using a hash function.

94. Write a C++ program to implement a binary search tree.

Answer:

```
#include <iostream>

struct Node {
    int data;
```

```
    Node* left;
    Node* right;
};

class BinarySearchTree {
public:
    Node* root;

    BinarySearchTree() {
        root = nullptr;
    }

    // Inserts a new value into the tree
    void insert(int data) {
        Node* newNode = new Node();
        newNode->data = data;
        newNode->left = nullptr;
        newNode->right = nullptr;

        if (root == nullptr) {
            root = newNode;
            return;
        }

        Node* current = root;
        while (true) {
            if (data < current->data) {
```

```
            if (current->left == nullptr) {
                current->left = newNode;
                break;
            } else {
                current = current->left;
            }
        } else {
            if (current->right == nullptr) {
                current->right = newNode;
                break;
            } else {
                current = current->right;
            }
        }
    }
};

int main() {
    BinarySearchTree tree;
    tree.insert(5);
    tree.insert(3);
    tree.insert(7);
    tree.insert(2);
    tree.insert(4);
    tree.insert(6);
    tree.insert(8);
```

```
    return 0;
}
```

95. What is a heap?

Answer: A heap is a hierarchical data structure that follows a First In First Out (FIFO) ordering.

96. What is the difference between a depth-first search and a breadth-first search?

Answer: A depth-first search is an algorithm that explores a graph by traversing down each branch as far as possible before backtracking, while a breadth-first search is an algorithm that explores a graph by exploring all the nodes at each level before moving on to the next level.

97. What is a dynamic programming algorithm?

Answer: A dynamic programming algorithm is an algorithm that solves complex problems by breaking them down into smaller subproblems and then combining the solutions to the subproblems.

98. What is a greedy algorithm?

Answer: A greedy algorithm is an algorithm that makes decisions based on the current state of the problem and doesn't look ahead to future states.

99. What is the Big O notation?

Answer: The Big O notation is a way of representing the complexity of an algorithm, with lower values indicating better performance.

100. Write a function in C++ to delete a node in a linked list given the reference to the node to be deleted.

Answer:

```
// Function to delete a node from a linked list
void deleteNode(Node* node)
{
   Node* temp = node->next;
   // Copy the data of next node to current node
   node->data = temp->data;
   // Unlink the next node
   node->next = temp->next;
   // Delete the next node
   delete temp;
}
```