Explore More

Subcription : Premium CDAC NOTES & MATERIAL @99

Contact to Join
Premium Group

Click to Join
Telegram Group

For More E-Notes

Join Our Community to stay Updated

TAP ON THE ICONS TO JOIN!

	codewitharrays.in freelance project available to buy contact on 8007592194	
SR.NO	Project NAME	Technology
1	Online E-Learning Platform Hub	React+Springboot+MySql
2	PG Mates / RoomSharing / Flat Mates	React+Springboot+MySql
3	Tour and Travel management System	React+Springboot+MySql
4	Election commition of India (online Voting System)	React+Springboot+MySql
5	HomeRental Booking System	React+Springboot+MySql
6	Event Management System	React+Springboot+MySql
7	Hotel Management System	React+Springboot+MySql
8	Agriculture web Project	React+Springboot+MySql
9	AirLine Reservation System / Flight booking System	React+Springboot+MySql
10	E-commerce web Project	React+Springboot+MySql
11	Hospital Management System	React+Springboot+MySql
12	E-RTO Driving licence portal	React+Springboot+MySql
13	Transpotation Services portal	React+Springboot+MySql
14	Courier Services Portal / Courier Management System	React+Springboot+MySql
15	Online Food Delivery Portal	React+Springboot+MySql
16	Muncipal Corporation Management	React+Springboot+MySql
17	Gym Management System	React+Springboot+MySql
18	Bike/Car ental System Portal	React+Springboot+MySql
19	CharityDonation web project	React+Springboot+MySql
20	Movie Booking System	React+Springboot+MySql

	freelance_Project available to buy contact on 8007592194	
21	Job Portal web project	React+Springboot+MySql
22	LIC Insurance Portal	React+Springboot+MySql
23	Employee Management System	React+Springboot+MySql
24	Payroll Management System	React+Springboot+MySql
25	RealEstate Property Project	React+Springboot+MySql
26	Marriage Hall Booking Project	React+Springboot+MySql
27	Online Student Management portal	React+Springboot+MySql
28	Resturant management System	React+Springboot+MySql
29	Solar Management Project	React+Springboot+MySql
30	OneStepService LinkLabourContractor	React+Springboot+MySql
31	Vehical Service Center Portal	React+Springboot+MySql
32	E-wallet Banking Project	React+Springboot+MySql
33	Blogg Application Project	React+Springboot+MySql
34	Car Parking booking Project	React+Springboot+MySql
35	OLA Cab Booking Portal	React+NextJs+Springboot+MySql
36	Society management Portal	React+Springboot+MySql
37	E-College Portal	React+Springboot+MySql
38	FoodWaste Management Donate System	React+Springboot+MySql
39	Sports Ground Booking	React+Springboot+MySql
40	BloodBank mangement System	React+Springboot+MySql

41	Bus Tickit Booking Project	React+Springboot+MySql
42	Fruite Delivery Project	React+Springboot+MySql
43	Woodworks Bed Shop	React+Springboot+MySql
44	Online Dairy Product sell Project	React+Springboot+MySql
45	Online E-Pharma medicine sell Project	React+Springboot+MySql
46	FarmerMarketplace Web Project	React+Springboot+MySql
47	Online Cloth Store Project	React+Springboot+MySql
48	Train Ticket Booking Project	React+Springboot+MySql
49	Quizz Application Project	JSP+Springboot+MySql
50	Hotel Room Booking Project	React+Springboot+MySql
51	Online Crime Reporting Portal Project	React+Springboot+MySql
52	Online Child Adoption Portal Project	React+Springboot+MySql
53	online Pizza Delivery System Project	React+Springboot+MySql
54	Online Social Complaint Portal Project	React+Springboot+MySql
55	Electric Vehical management system Project	React+Springboot+MySql
56	Online mess / Tiffin management System Project	React+Springboot+MySql
57		React+Springboot+MySql
58		React+Springboot+MySql
59		React+Springboot+MySql
60		React+Springboot+MySql

Spring Boot + React JS + MySQL Project List

Sr.No	Project Name	YouTube Link
1	Online E-Learning Hub Platform Project	https://youtu.be/KMjyBaWmgzg?si=YckHuNzs7eC84-IW
2	PG Mate / Room sharing/Flat sharing	https://youtu.be/4P9clHg3wvk?si=4uEsi0962CG6Xodp
3	Tour and Travel System Project Version 1.0	https://youtu.be/-UHOBywHaP8?si=KHHfE_A0uv725f12
4	Marriage Hall Booking	https://youtu.be/VXz0kZQi5to?si=llOS-QG3TpAFP5k7
5	Ecommerce Shopping project	https://youtu.be/vJ_C6LkhrZ0?si=YhcBylSErvdn7paq
6	Bike Rental System Project	https://youtu.be/FIzsAmIBCbk?si=7ujQTJqEgkQ8ju2H
7	Multi-Restaurant management system	https://youtu.be/pvV-pM2Jf3s?si=PgvnT-yFc8ktrDxB
8	Hospital management system Project	https://youtu.be/IynIouBZvY4?si=CXzQs3BsRkjKhZCw
9	Municipal Corporation system Project	https://youtu.be/cVMx9NVyI4I?si=qX0oQt-GT-LR_5jF
10	Tour and Travel System Project version 2.0	https://youtu.be/_4u0mB9mHXE?si=gDiAhKBowi2gNUKZ

Sr.No	Project Name	YouTube Link
11	Tour and Travel System Project version 3.0	https://youtu.be/Dm7nOdpasWg?si=P_Lh2gcOFhlyudug
12	Gym Management system Project	https://youtu.be/J8_7Zrkg7ag?si=LcxV51ynfUB7OptX
13	Online Driving License system Project	https://youtu.be/3yRzsMs8TLE?si=JRI_z4FDx4Gmt7fn
14	Online Flight Booking system Project	https://youtu.be/m755rOwdk8U?si=HURvAY2VnizIyJlh
15	Employee management system project	https://youtu.be/lD1iE3W_GRw?si=Y_jv1xV_BljhrD0H
16	Online student school or college portal	https://youtu.be/4A25aEKfei0?si=RoVgZtxMk9TPdQvD
17	Online movie booking system project	https://youtu.be/Lfjv_U74SC4?si=fiDvrhhrjb4KSlSm
18	Online Pizza Delivery system project	https://youtu.be/Tp3izreZ458?si=8eWAOzA8SVdNwlyM
19	Online Crime Reporting system Project	https://youtu.be/0UlzReSk9tQ?si=6vN0e70TVY1GOwPO
20	Online Children Adoption Project	https://youtu.be/3T5HC2HKyT4?si=bntP78niYH802I7N

1. There are how many ways for delimiting an un-indexed sequence of objects?

Answers

1. 1

2. 2

3. 3

4. 4

2. What is correct output of given code snippets?

```
#include <iostream>
#include <fstream>

using namespace std;

int main()
{
    fstream fs;

    char data[16];

    fs.open("abc.txt");

    fs << "hello world" << endl;

    getline(fs, data);

    fs.close();

    cout << data << endl;

    return 0;
}
```

Answers

1. hello world

2. syntax error

3. no output

4. runtime error

3. What is correct output of given code snippets?

```
#include <iostream>
#include <fstream>

using namespace std;

int main()
{
    ofstream ofs;

    ofs.open("abc.txt");

    ofs << "hello world" << endl;

    cout << ofs.tellp() << endl;
    ofs.close();

    return 0;
}
```

Answers

1. 11

2. 12

3. 13

4. Syntax error

4. Which from the following format flag is not included in ios: : adjustfield mask?

Answers

1. ios : :left

2. ios : :right

3. ios : :oct

4. ios : :internal

5. What is the output of this program?

```
#include <iostream>
 using namespace std;
 class vec
 {
     public:
     vec(float f1, float f2)
     {
         x = f1;
         y = f2;
     }
     vec()
     {
     }
     float x;
     float y;
 };
 vec addvectors(vec v1, vec v2);
 int main()
 {
     vec v1(3, 6);
     vec v2(2, -2);
     vec v3 = addvectors(v1, v2);
     cout << v3.x << ", " << v3.y << endl;
 }
 vec addvectors(vec v1, vec v2)
 {
     vec result;
     result.x = v1.x + v2.x;
     result.y = v1.y + v2.y;
     return result;
 };
```

```
    return result;
};
```

Answers

1. 4, 5

2. 4, 4

3. 5, 4

4. 5, 5

6. The I/O operations that use the extraction and insertion operators, are called

Answers

1. Formatted I/O

2. Formatted strings

3. Formatted flags

4. All of them

7. In C++, if a class has a private constructor and a friend class wants to create instances of that class, which of the following is the correct approach?

Answers

1. The friend class should declare a public constructor in the target class

2. The friend class should declare a private constructor in the target class

3. The friend class should use a static member function in the target class to create instances

4. It is not possible for a friend class to create instances of a class with a private constructor

8. class DAC {
public:
    virtual void java();
};

class DBDA : public DAC {
public:
    void java() ;
};

Answers

1. The java function in class DBDA is not allowed to override the java function in class DAC

2. The java function in class DBDA must have the same access specifier as the java function in class DAC

3. The java function in class DBDA hides the java function in class DAC

4. The java function in class DBDA must have the same signature as the java function in class DAC

9. In C++, when a derived class has multiple base classes and these base classes have a common ancestor, what issue may arise?

Answers

1. Diamond problem

2. Segmentation fault

3. Stack overflow

4. Compile-time error

10. Which of the following statements is true about the default constructor generated by the compiler?

Answers

1. The default constructor is generated only if there are no other constructors in the class

2. The default constructor is always generated, even if there are other constructors in the class

3. The default constructor is generated only if it is explicitly declared in the class

4. The default constructor is generated only for classes with virtual functions

11.
```
#include <iostream>
class A {
public:
    A() { std::cout << "A"; }
    A(const A&) { std::cout << "B"; }
    virtual ~A() { std::cout << "C"; }
};

class B : public A {
public:
    B() { std::cout << "D"; }
    B(const B& other) : A(other) { std::cout << "E"; }
    ~B()  { std::cout << "F"; }
};

int main() {
    A* obj = new B();
    delete obj;
    return 0;
}
```
What will be the output of the above C++ code?

Answers

1. ADBCF

2. ABCDEF

3. ADFC

4. ADBFEC

12. In C++, what is the significance of declaring a base class function as pure virtual and providing its definition within the base class?

Answers

1. It allows the function to be overridden in derived classes

2. It enforces the derived classes to provide their own definition

3. It allows the creation of an object of the base class

4. It makes the function non-overridable

13.
```
class Base {
protected:
    virtual void func() {
        cout << "Base";
    }
};

class Derived : public Base {
protected:
    void func()  {
        cout << "Derived";
    }
};
```
Which of the following statements is true regarding the function func() in the derived class?

Answers

1. It is inaccessible outside the derived class using derived class object

2. It is public in the derived class

3. It is private in the derived class

4. It is accessible in the derived class but cannot be overridden

14. _____ is used as a conditional operator?

Answers

1. ? :

2. > >

3. : :

4. / *

15. Consider the following two pieces of codes and choose the best answer

Code 1:

```
switch (x)
{
case 1:
    cout <<"m is 1";
    break;
case 2:
    cout <<"m is 2";
    break;
default:
    cout <<"value of m unknown";
}
```

Code 2:

```
if(x == 1)
{
    Cout << "m is 1";
}
else if (x == 2)
{
    Cout << "m is 2";
}
else
{
    Cout << "value of m unknown";
}
```


Answers

1. The first code produces more results than second

2. Both of the above code fragments produce different effects

3. Both of the above code fragments have the same behaviour

4. The second code produces more results than first.

16. Which of the following operator can replace a simple if-else construct?

Answers

1. Unary operator

2. Ternary operator

3. Assignment operator

4. Arithmetic operator

17. Find the output of below program:-

```
int main()
{
    int i = 0, a = 0;
    do
    {
        if (i % 5 == 0)
        {
            std::cout << a;
            a++;
        }
        ++i;
    } while (i < 10);
    cout << a;
    return 0;
}
```

Answers

1. 0

2. 01

3. 012

4. 0123

18. Which of the following is a properly defined structure?

Answers

1. struct {int a;}

2. struct a_struct int a;

3. struct a_struct {int a;}

4. struct a_struct {int a;};

19. What will be the output of this program?

```
int main() {
    int arr[2][3] = {{1, 2, 3}, {4, 5, 6}};
    int rows = sizeof(arr) / sizeof(arr[0]);
    int cols = sizeof(arr[0]) / sizeof(arr[0][0]);
    for(int i = rows - 1; i >= 0; i--) {
        for(int j = cols - 1; j >= 0; j--) {
            cout << arr[i][j] << " ";
        }
        cout << endl;
    }
    return 0;
}
```


Answers

1. 1 2 3\n4 5 6

2. 3 2 1\n6 5 4

3. 6 5 4\n3 2 1

4. 4 5 6\n1 2 3

20. What will be the output of cout << 7/2;?

Answers

1. 3

2. 3.0

3. 3.5

4. 7/2

21. what are the following are advantages of templates?

Answers

1. Using templates we can reduce the execution time.

2. Using templates we can reduce code size

3. Using templates we can reduce developers efforts.

4. Both A and C

22. What will the output?

```
#include <iostream>
using namespace std;

template <class T>
class Test
{
private:
    T num;
public:
    static int count;
    Test()  {   count++;   }
};

template<class T>
int Test<T>::count = 0;

int main()
{
    Test<int> a;
    Test<int> b;
    Test<double> c;
    cout << Test<int>::count   << endl;
    cout << Test<double>::count << endl;
    return 0;
}
```


Answers

1. 0 0

2. 2 2

3. 2 1

4. 1 0

23. Which of the following operators we need to use for having access to type_info class object during RTTI?

Answers

1. typeid

2. sizeof

3. reinterpret_cast

4. dynamic_cast

24. Which algorithm is used to find the maximum element in a range in STL?

Answers

1. max_element()

2. find_max()

3. maximum()

4. find_maximum()

25. Which STL container provides a dynamic array implementation with constant time complexity for random access and insertion/deletion at the end?

Answers

1. list

2. vector

3. deque

4. set

26. What does the following code snippet do?

```
#include <iostream>
#include <vector>
#include <algorithm>

int main() {

    std::vector<int> numbers = {5, 2, 8, 1, 7};
    numbers.erase(std::remove(numbers.begin(), numbers.end(), numbers.front()), numbers.end());
  }
```

Answers

1. Sort the vector in ascending order.

2. Remove the first element from the vector.

3. Find average.

4. Reverse the vector

27. What happens if a pure virtual function is not implemented in a derived class?

Answers

1. Compile-time error

2. Link-time error

3. Runtime error

4. No error

28. What is the purpose of the following C++ code snippet?

```
#include <iostream>

class Base {
public:
    virtual void print() const = 0;
};

class Derived : public Base {
public:
    void print() const override { std::cout << "Derived" << std::endl; }
};

int main() {
    Base* obj = new Derived();
    obj->print();
    delete obj;
    return 0;
}
```

Answers

1. Implements an interface in C++

2. Defines a pure virtual function

3. Creates an abstract base class

4. Enforces encapsulation

29. What is the output of the following C++ code snippet?

```
#include <iostream>

class A {
public:
    A() { std::cout << "A"; }
    ~A() { std::cout << "~A"; }
};

class B : public A {
public:
    B() { std::cout << "B"; }
    ~B() { std::cout << "~B"; }
};

int main() {
    A* obj = new B();
    delete obj;
    return 0;
}
```

Answers

1. AB~BA

2. BA~A

3. AB~A

4. BA~BA

30. What is the role of the pure specifier in C++ when used with a virtual function in an abstract class?

Answers

1. It ensures the function is implemented in the abstract class.

2. It indicates that the function is pure virtual.

3. It specifies that the function cannot be overridden.

4. It is not a valid specifier in this context.

31. When is the virtual destructor in an abstract class particularly useful?

Answers

1. It is never useful in an abstract class

2. When objects of the abstract class are dynamically allocated

3. When the abstract class is a base class for polymorphism

4. When the abstract class has no derived classes

32. What is the output of the following C++ code snippet?

```
#include <iostream>

int main() {
    try {
        try {
            throw "Inner exception";
        } catch (const char* msg) {
            std::cout << "Inner Catch: " << msg << std::endl;
            throw; // What does this line do?
        }
    } catch (...) {
        std::cout << "Outer Catch" << std::endl;
    }
    return 0;
}
```

Answers

1. Inner Catch: Inner exception

2. Outer Catch

3. Inner Catch: Inner exception followed by Outer Catch

4. Compiler error

33. In C++, when should you catch exceptions by reference rather than by value?

Answers

1. Always

2. When dealing with built-in types

3. When the exception hierarchy is well-defined

4. When catching polymorphic exceptions

34. What will be the output ?

```
#include<iostream.h>
enum week{Mon, Tue, Wed, Thur, Fri, Sat, Sun};
int main()
{
    enum week day;
    day = Wed;
    cout<<day;
    return 0;
}
```

Answers

1. 2

2. 1

3. 3

4. 0

35. An inline function is expanded during _____

Answers

1. compile-time

2. run-time

3. never expanded

4. end of the program

36. What will be the output of the following C++ code?

```
#include <iostream>
using namespace std;
int fun(int x = 0, int y = 0, int z){
    return (x + y + z);
}
int main() {

    cout << fun(10);

    return 0;
}
```

Answers

1. 10

2. 0

3. Error

4. Segmentation fault

37. What will be the output ?

```
void square(int *p){
 int a = 10;
 p = &a;
 *p = (*p) * (*p);
}

int main(){
 int a = 10;
 square(&a);
 cout << a << endl;
}
```

Answers

1. 10

2. 00

3. Error

4. Segmentation fault

38. How to create a dynamic array of pointers (to integers) of size 10 using new in C++?

Answers

1. int *arr = new int *[10];

2. int **arr = new int *[10];

3. int *arr = new int [10];

4. int *arr = new int [10];

39. If new operator is used, when is the constructor called?

Answers

1. Depends on code

2. Before the allocation of memory

3. After the allocation of memory

4. Constructor is called to allocate memory

40. Which one of the following is the correct definition of the "is_array();" function in C++?

Answers

1. It checks that the specified variable is of the array or not

2. It checks that the specified array of single dimension or not

3. It checks that the array specified of multi-dimension or not

4. Both B and C

https://www.youtube.com/@codewitharrays

https://www.instagram.com/codewitharrays/

https://t.me/codewitharrays Group Link: https://t.me/ccee2025notes

+91 8007592194 +91 9284926333

codewitharrays@gmail.com

https://codewitharrays.in/project