1. Which of the following statement is incorrect in context of the default argument?

1.Including main function, we can assign default argument to parameters of any global function as well member function.
2.Default arguments are always assigned from left to right direction.
3.In case of modular approach, the default argument must appear in the definition part.

Answers

1. Only 1 is incorrect

2. Both 1 and 2 are incorrect

3. Both 2 and 3 are incorrect

4. All the above

2. Which of the following statements are true in context of the program below?

```
#include<iostream>
using namespace std;
class Point
{
private:
        int xPos;
        int yPos;
public:
        Point(void)
        {
                this->xPos=0;
                this->yPos=0;
        }
        Point(int value)
        {
                this->xPos=value;
                this->yPos=value;
        }

        Point(int x,int y)
        {
                this->xPos=x;
                this->yPos=y;
        }
};
```

```
int main()
{
        Point(10,20);
        Point pt2=(30,40);
        Point pt3();

        return 0;
}
```

Answers

1. point(10,20); is the anonymous object

2. Compile time error

3. For pt2 object single parameterized constructor will call and xPos and yPos will initialize by 30

4. For pt2 object single parameterized constructor will call and xPos and yPos will initialize by 40

5. Both A and D

3. Which of the following statements are true in context of the program below?

```
#include<iostream>
using namespace std;
class Employee
{
        string name;
        int empid;
        float salary;
    public :
        Employee(string name,int empid,float salary)
        {
                this->name=name;
                this->empid=empid;
                this->salary=salary;
        }
        void print_emp_info()
        {
                cout<<name<<endl;
                cout<<empid<<endl;
                cout<<salary<<endl;
        }
};
int main()
{
  Employee emp("John",201,15000);
  emp.print_emp_info();
  return 0;
}
```

Answers

1. ("John",201,15000) are representing state of emp object

2. print_emp_info() is representing behaviour of object emp.

3. name, empid and salary are representing state of emp object

4. Both A and B are correct

4. Which of the following statement is incorrect in context of inline function?

1.Inline functions gets process during compile time
2.If we implement member function inside class then by default it is considered as inline
3.If function is inline we cannot separate its code in multiple files.
4.If we want to reduce overhead to the compiler during function call then we should declare function as a inline

Answers

1. Only 1 is incorrect

2. Both 1 and 2 are incorrect

3. Both 1 and 3 are incorrect

4. 1, 2 and 4 are incorrect

5. None of the above

5. Which among the following is correct?

Answers

1. Private specifier must be used before public specifier

2. Private specifier must be used before protected specifier

3. Private specifier must be used first

4. Private specifier can be used anywhere in class

6. Which of the following structure syntax is incorrect in case of c++?

1.
```
struct
{
     char name[ 30 ];
     int empid;
     float salary;
 public :
     void print_emp_info()
     {
         cout<<name<<endl;
         cout<<empid<<endl;
         cout<<salary<<endl;
     }
}emp;
int main()
{
    strcpy(emp.name,"John");
    emp.empid=101;
    emp.salary=10000;
    emp.print_emp_info();
}
```

2.
```
typedef struct
{
     char name[ 30 ];
     int empid;
     float salary;
 public :
     void print_emp_info()
     {
         cout<<name<<endl;
         cout<<empid<<endl;
         cout<<salary<<endl;
     }
}emp;
int main()
{
    emp e1;
    strcpy(e1.name,"John");
    e1.empid=101;
    e1.salary=10000;
    e1.print_emp_info();
}
```

3.
```
typedef struct
{
 private:
     char name[ 30 ];
     int empid;
     float salary;
 public :
     void print_emp_info()
     {
         cout<<name<<endl;
         cout<<empid<<endl;
         cout<<salary<<endl;
     }
}emp;
int main()
{
    emp e1;
    strcpy(e1.name,"John");
    e1.empid=101;
    e1.salary=10000;
    e1.print_emp_info();
}
```

Answers

1. All 1, 2 and 3 are incorrect

2. Only 3 is incorrect

3. Only 1 is incorrect

4. Both 1 and 2 are incorrect

```
7. #include <iostream>
using namespace std;
namespace na
{
int num1 = 10;
}
int main( void )
{
int num1 = 20;
using namespace na;
cout<<"Num1 : "<<num1<<endl;
Cout<<"Num1 :"<<na::num1<<endl;
return 0;
}
```

Answers

1. 20 20

2. 10 10

3. 20 10

4. Error

5. None of the above

```
8. What is  output of this program?
void fun(int, int);

int main()
{
        fun(1,2);
        return 0;
}

void fun(int x,int y,int z)
{
        Cout<<endl<<x<<endl<<y<<endl<<z;
}


void fun(int x,int y)
{
        cout<<endl<<x<<endl<<y;
}
```

Answers

1. 1 2

2. Compile time error

3. 1 2

4. none of the above

9. Which of following access specifier is used in a structure definition by default?

Answers

1. protected

2. public

3. private

4. friend

10. Which of the following statements is/are correct in context of class?

1.Class is a collection of data member and member functions.
2.Class is a physical entity.

Answers

1. Only 2 is correct

2. Only 1 is correct

3. Both 1 and 2 are correct

4. None of the above

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

```
#include<iostream>
using namespace std;
int main( void )
{
        int num1 = 10;
        int num2 = 20;
        int &num3 = num1;
        num3 = num2;
        ++ num2;
        cout<<"Num1 : "<<num1<<endl;
        cout<<"Num2 : "<<num2<<endl;
        cout<<"Num3 : "<<num3<<endl;
        return 0;
}
```

Answers

1. Num1 : 10 Num2 : 21 Num3 : 10

2. Compile time Error

3. Num1 : 20 Num2 : 21 Num3 : 20

4. Num1 : 10 Num2 : 21 Num3 : 20

Question 2 of 10

Which of the following statements is correct?

1.A reference is useful, when we intend to change the values of actual arguments through the function call.
2.A reference is useful, when we want to save memory by preventing the ceation of large structure variable that are being passed to the function.

- ○ Only 1 is correct.
- ○ Only 2 is correct.
- ● Both 1 and 2 are correct.
- ○ Both 1 and 2 are incorrect.

3. During dynamic memory allocation in C++, what happens if a new operator fails to allocate memory?

Answers

1. It returns False

2. It returns NULL

3. Throws bad_alloc exception

4. None of these

Question 4 of 10

What is output of following code?

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

int main()
{
        int &p;
        int i=5;
cout<<"i :"<<i<<endl;
        &p=a;
p++;
cout<<"i :"<<i<<endl;
        return 0;
}
```

- ○ 5 5
- ○ 5 6
- ● compile time error
- ○ 5 6

5. Which of the following statements are correct in context of below code?

```
#include<iostream>
using namespace std;
int main( void )
{
   int *ptr = new int(3);
   return 0;
}
```

Answers

1. we are allocating memory for a single variable but memory will be initialized with value 3.

2. we are allocating memory for array of three integer variable

3. we are allocating memory for array of three integer variable but memory will be initialized with 0 value

4. None of the above

6. What is output of following code?

```
using namespace std;
#include<iostream>
int main()
{
        int i;
        int arr[4]={10,20,30,40};
        int (&k)[4]=arr;
        for(i=0;i<4;i++)
        cout<<k[i]<<" ";

        return 0;
}
```

Answers

1. 10 20 30 40

2. 10 20

3. compile time error

4. none of the above

7. Which of the following statements are correct in context of below code in C++?

```
#include<iostream>
#include<stdlib.h>
using namespace std;
int main( void )
{
    int *p = malloc(10);
}
```

Answers

1. It will allocate memory for 10 integer variables in the heap section consecutively.

2. Compile time error

3. It will allocate memory for one integer variable in the heap section and memory will be initialized with 10 value.

4. Runtime error

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

```
#include<iostream>
#include<stdlib.h>
using namespace std;
class Test
{
        public:
                Test()
                {
                        cout << "Constructor called";
                }
};
int main()
{
        Test *t = (Test *) malloc(sizeof(Test));
    return 0;
}
```

Answers

1. Constructor called

2. Constructor will not be called

3. Compiler Error

4. Runtime error

9. Which of the following statements are incorrect related to malloc() function?

Answers

1. We cannot initialize memory with user defined value using malloc()

2. If we create object using malloc() then constructor do not get call

3. Using malloc() we can reserved space on heap segment only

4. none of the above

10. Which of the following statements is/are incorrect about references in C++?
1.Once reference is initialized, we can change its referent later.
2.Reference is an alias or another name given to the existing memory location/object.

Answers

1. Both 1 and 2 are incorrect

2. Only 1 is incorrect

3. Only 2 is incorrect

4. None of the above

<CODEWITHARRAY'S/>

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

```
#include <iostream>
using namespace std;
class Point
{
    int x, y;
 public:
   Point(const Point &p)
   {
           x = p.x;
           y = p.y;
   }
   int getX()
   {
           return x;
   }
   int getY()
   {
           return y;
   }
};

int main()
{
    Point p1;
    Point p2 = p1;
    cout << "x = " << p2.getX() << " y = " << p2.getY();
    return 0;
}
```

Answers

1. x = garbage value y = garbage value

2. x = 0 y = 0

3. Compiler Error

4. None of the above

2. _____ is called when an already created object is assigned to another already created object.

Answers

1. Copy Constructor

2. Default Constructor

3. Assignment operator

4. None of above

3. Which of the following is/are automatically added to every class, if the programmer does not write it in the class?

Answers

1. Copy Constructor
2. Assignment Operator
3. Both A and B
4. None of the above

4. _____ is called when a new object is created from an existing object, as a copy of the existing object

Answers

1. Copy Constructor
2. Default Constructor
3. Assignment operator
4. None of above

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

```
#include<iostream>
using namespace std;
class Point
{
    int x;
public:
    Point(int x)
        {
        this->x = x;
        }
    Point(const Point p)
    {
        x = p.x;
    }
    int getX()
    {
        return x;
    }
};
int main()
{
   Point p1(10);
   Point p2 = p1;
   cout << p2.getX();
   return 0;
}
```

Answers

1. Compiler Error: p must be passed by reference
2. Garbage Value
3. 10.
4. None of the above

6. When a copy constructor may be called?

Answers

1. When an object of the class is returned by value.

2. When an object of the class is passed (to a function) by value as an argument.

3. When an already created object is assigned to newly created object of the same class

4. When the compiler generates a temporary object.

5. All of the above

7. Which of the below is correct syntax to call copy constructor?

Answers

1. Employee e1 , e2;

2. Employee e1 = e2;

3. Employee e1 , e2; e1=e2

4. none of above

8. If a copy constructor is not defined in a class then

Answers

1. Compiler itself defines one

2. Conpiler gives Syntax Error

3. Compiler gives Run time Error

4. None of Above

9. Which of the below is correct syntax to call an assignment operator?

Answers

1. Employee e1 , e2;

2. Employee e1 = e2;

3. Employee e1 , e2; e1=e2

4. None of the above

10. Default copy constructor does _____

Answers

1. Deep Copy

2. Shallow copy

3. Both A and B

4. None of Above

1. In case of operator overloading, operator function must be _____
    1. Static member functions
    2. Non- static member functions
    3. Friend Functions

Answers

1. Only 2

2. Only 1,3

3. Only 2,3

4. All 1,2,3

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

```
#include <iostream>
using namespace std;
class Test
{
        public:
        Test();
};
Test::Test()
{
    cout << " Constructor Called. ";
}
void fun()
{
        static Test t1;
}
int main()
{
    cout << " Before fun() called. ";
    fun();
    fun();
    cout << " After fun() called. ";
    return 0;
}
```

Answers

1. Constructor Called. Before fun() called. After fun() called.

2. Before fun() called. Constructor Called. Constructor Called. After fun() called.

3. Before fun() called. Constructor Called. After fun() called.

4. Constructor Called. Constructor Called. After fun() called. Before fun() called.

3. Which of the following operator(s) cannot be overloaded?

Answers

1. . (Member Access or Dot operator)

2. ?: (Ternary or Conditional Operator)

3. :: (Scope Resolution Operator)

4. All of the above

4. Which of the below mentioned can be made constant

Answers

1. Member functions of a class

2. Function arguments

3. Class data member

4. All of Above

5. When overloading unary operators using Friend function,it requires_____ argument/s.

Answers

1. Zero

2. One

3. Two

4. none of above

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

```
#include <iostream>
using namespace std;
class Point
{
    int x, y;
        public:
    Point(int i = 0, int j =0)
        {
        x = i;
        y = j;
        }
    int getX() const
    {
        return x;
    }
    int getY()
    {
       return y;
    }
};
int main()
{
    const Point t;
    cout << t.getX() << " ";
    cout << t.gety();
    return 0;
}
```

Answers

1. Garbage Values

2. 0 0

3. Compiler Error in line cout << t.getX() << " ";

4. Compiler Error in line cout << t.gety();

7. Which of the following statements is/are incorrect about static data member function in C++?

1. Static data member gets space inside object
2. If we want to share value of the data member in all the objects of same class then we should declare data member static

Answers

1. Only 2 is incorrect

2. Only 1 is incorrect

3. Both 1 and 2 are incorrect

4. None of the above

8. If we want to overload, binary operator using member function then operator function should take.....parameter.

Answers

1. Zero
2. One
3. Two
4. none of above

9. Syntax for making member function as Constant is _____.

Answers

1. void display()
2. void display() const
3. const void display()
4. void const display()

10. Which keyword must be used to declare a member function as a constant member function?

Answers

1. Constant
2. Const
3. FunctionConst
4. Unchanged

1. What is meant by template parameter?

Answers

1. It can be used to pass a type as argument
2. It can be used to evaluate a type
3. It can of no return type
4. None of the mentioned

2. If we extend abstract class then it is _____ to override pure virtual function in derived class otherwise derived class can be considered as abstract?

Answers

1. Mandatory
2. Optional
3. Depends on User Requirement
4. None

3. Why does diamond problem arise due to multiple inheritance?

Answers

1. Methods with same name creates ambiguity and conflict
2. Derived class can't distinguish the owner class of any derived method
3. Derived class gets overloaded with more than two class methods
4. None of above

4. Derived Class object can be treated as Base class object this concept is called as?

Answers

1. Inheritance
2. Upcasting
3. Downcasting
4. None

5. The process of finding type(data type/ class name) of object/variable at runtime is called?

Answers

1. Runtime Type Information
2. Runtime Type Identification
3. Both A and B
4. Function Overloading

6. Which of the following member function that is declared within a base class and redefined by derived class?

Answers

1. Static function
2. Friend function
3. Virtual function
4. None

7. To which type of class, we can apply RTTI?

Answers

1. Encapsulation
2. Polymorphic
3. Derived
4. Inherited

8. If we call any virtual/non virtual function on object then it is considered as?

Answers

1. Late Binding.
2. Early Binding
3. Virtual Binding
4. None

9. Which of the following keyword is used to avoid diamond problem?

Answers

1. Static
2. Virtual
3. New
4. None

10. allows us to treat a derived type as though it were its base type

Answers

1. Deep Copy
2. Constant data members
3. Upcasting
4. Downcasting

1. Analysis process which involves analyzing and designing of system from an object oriented programming is termed as

Answers

1. Object analysis
2. Object oriented Analysis
3. Overall oriented analysis
4. System analysis

2. When a protected member is inherited in public mode, it becomes ……. in the derived class too and therefore is accessible by member function of the derived class.

Answers

1. protected
2. private
3. public
4. friend

3. Which feature of OOPS illustrated the code reusability?

Answers

1. Polymorphism
2. Abstraction
3. Encapsulation
4. Inheritance

4. function that can perform different type of tasks, where the function name remains same, which feature of OOP is used here?

Answers

1. Inheritance
2. Polymorphism
3. Abstraction
4. Constructor

5. If single base class is having multiple derived classes then such inheritance is called?

Answers

1. Multiple Inheritance
2. Hierarchical Inheritance
3. Single Inheritance
4. Multilevel Inheritance

6. Default mode of inheritance is

Answers

1. Private

2. Public

3. Protected

4. None

7. Procedural programming is also referred to as _____

Answers

1. Routine Programming

2. Object oriented programming

3. Declarative programming

4. None of above

8. What is the output of this program?

```
#include<iostream>
using namespace std;
class Person
{
    public:
 Person()
    {
     cout <<"We are in Person class"<< endl;
    }
};

class Student: public Person
{

};

class Faculty: public Person
{

};

int main()
{
     Student student;
     Faculty faculty;
}
```

Answers

1. We are in Person class

2. We are in Person class We are in Person class

3. Compile time error

9. Which of the following Data members of base class inherits into derived class?

Answers

1. Public

2. Private , Protected

3. static and nonstatic

4. All of the above

10. In POP Programs are divided into?

Answers

1. Classes

2. Functions

3. Both A and B

4. None

1. If you want to write multiple functions in a class with same name, then what C++ feature will you use?

Answers

1. Function overriding

2. Encapsulation

3. Function overloading

4. None

2. Correct way of creating an object of a class called Student is

Answers

1. Student student;

2. Student *student=new Student();

3. Only B

4. A & B Both

3. Copy constructor must recieve its arguments by

Answers

1. only pass-by-refernece

2. only pass-by-value

3. only pass by address

4. All of the above

4. Class xyz: public abc,public mnp{}; this is called as

Answers

1. Multilevel inheritance

2. Multiple inheritance

3. Hybrid inheritance

4. Hierarchical Inheritance

5. What are the different type of inheritance

Answers

1. multiple

2. Multi-level

3. Single level

4. All of the above

6. Which of the following is an abstract data type?

Answers

1. int

2. boolean

3. bool

4. class

7. Friend function speically is used for

Answers

1. to access private data memeber of different class

2. to avoid private data memeber of different class

3. Both A & B

4. None of the above

8. What is purpose of Virtual destructor?

Answers

1. Virtual destructor is to destroy the VTable

2. Destructor can be virtual so that we can override the destructor of base class in derived class.

3. To maintain the call hierarchy of destructors of base and derived classes.

4. None

9. Which of the following can not be declare as virtual in C++?

Answers

1. constructor

2. destructor

3. function

4. friend function

10. C++ abstract class can contain

Answers

1. Pure virtual function

2. Non-virtual function

3. Only pure virtual function

4. pure virtual,non-virtual function,virtual function

11. Exposing only necessary information to usersis know as ...

Answers

1. Abstraction

2. Encapsulation

3. Data hiding

4. Composition

12. IS-A relationship in C++ is

Answers

1. Inheritance

2. Encapsulation

3. Composition

4. None

13. cout is a/an

Answers

1. operator

2. function

3. object

4. macro

14. To Create object of class Employee on stack which of the following syntax is correct.

Answers

1. Employee *emp=new Employee();

2. Employee emp=new Employee();

3. Employee emp;

4. None

15. The function which is called without object name of class is Called as .....

Answers

1. constant member function

2. Inline function

3. Static member function

4. Friend function

16. By which class we can create only single object through out programe life

Answers

1. Abstract Class

2. Friend Class

3. Singleton Class

4. All Classes

17. Inheritance is used for achieving ...

Answers

1. Class Re-usability

2. Creating a hierarchy of classes

3. Extendibility

4. All of Above

18. Following keyword is used before a function in a base class to be overridden in derived class in C++

Answers

1. override

2. virtual

3. new override

4. new

19. When we create object like(e.g Student *student =new Student();) then it is stored object on

Answers

1. Stack

2. Heap

3. Both A & B

4. None of the above

20. Which of the following can be overloaded?

Answers

1. Object

2. Operators

3. Functions

4. Both B and C