1. What is the output of the following code?

```
int num1 = 10;
namespace Outer
{
        int num1 = 20;
}
int main( void )
{
        int num1 = 30;
        using namespace Outer;
        cout<<"Num1 :         "<<num1<<endl;
        return 0;
}
```

Answers
1. 10
2. 20
3. **30**
4. Compilation Error

2.
```
#include<iostream>
using namespace std;
#define INC(X) X++
int main()
{
int X = 4;
cout<<INC( X++ ) <<endl;
return 0;
}
```
Answers
1. 4
2. 5
3. 6
4. Compilation error.

3. What is the output of the following code?

```
class Program
{
public:
        void print( int num1 )
        {
                cout<<"Instance:"<<num1<<endl;
        }
        static void print( int num1 )
        {
                cout<<"static:"<<num1<<endl;
        }
};
int main( void )
```

```
{
        Test test;
        test.print(10);
}
```
Answers
1. Instance: 10
2. Static: 10
3. Error: static methods can not be called on object.
4. **Error: we can not overload static methods**

4. What will be the output of the following program?

```
int main( void )
{
        int num1 = 10;
        int& num2 = num1;
        num2 = num1 ++;
        num1 = num2 ++;
        cout<<num1<<"        "<<num2<<endl;
        return 0;
}
```
Answers
1. **10 10**
2. 11 11
3. 12 12
4. 11 12

5. If X is name of the class then what is the correct way to declare copy constructor of X?
Answers
1. X(const X other)
2. X(const X* other)
3. X(const &X other)
4. **X(const X& other)**

6. Which one of the following statements is not true about destructor?
Answers
1. We can declare destructor inline.
2. **We can't declare destructor virtual**
3. We can call destructor explicitly
4. We can't overload destructor

7. Which one of the following does not inherit into the derived class?
Answers
1. private members
2. static members
3. friend function
4. All of the above

8. _____ is the process of compartmentalizing the elements of an abstraction that constitute its structure and behavior?
Answers
1. Abstraction

2. **Encapsulation**
3. Hierarchy
4. Modularity

9. What is the output of the following program?

```
class Base
{
private:
	int num1;
private:
	Base( int num1 = 0)
	{
		this->num1 = num1;
	}
public:
	void print( void )
	{
		cout<<"Num1 :		"<<this->num1<<endl;
	}
	friend class Derived;
};
class Derived: public Base
{
private:
	int num2;
public:
	Derived( int num1, int num2 ) : Base( num2 )
	{
		this->num2 = num2;
	}
	void print( void )
	{
		Base::print();
		cout<<"Num2 :		"<<this->num2<<endl;
	}
};
int main( void )
{
	Derived derived( 10, 20 );
	derived.print();
	return 0;
}
```

Answers
1. 0 20
2. 10 20
3. 20 20
4. Error: Base class constructor is private

10. Which one of the following operators can not be overloaded as a non-member function?
Answers

1. <<
2. >>
3. ->
4. None of the above

11. Find the output of the following program?

```
class Base
{
public:
	virtual void print( void )
	{
		cout<<"Base::print"<<endl;
	}
};
class Derived : Base
{
public:
	virtual void print( void )
	{
		cout<<"Derived::print"<<endl;
	}
};
int main( void )
{
	Base* ptrBase = new Derived();
	ptrBase->print();
	delete ptrBase;
	return 0;
}
```

Answers
1. Base::print
2. Derived::print
3. **Compilation error due to upacasting**
4. Runtime error due to "delete ptrBase"

12. Observe the code and choose the correct option:

```
class Base
{
public:
	Base()
	{	}
	virtual void print( void )
	{
		cout<<"Base::print"<<endl;
	}
};
class Derived: public Base
{
public:
	Derived()
```

```
	{
		this->print();
	}
	virtual void print( void )
	{
		cout<<"Derived::print"<<endl;
	}
};
int main( void )
{
	Derived derived;
	return 0;
}
```

Answers

1. Due to static binding Base class's print function will call
2. **Due to static binding Derived class's print function will call**
3. Due to dynamic binding Derived class's print function will call
4. Nothing will print

13. What is the output of the following code?

```
int calculate( int num1, int num2 )throw(const char*)
{
	if( num2 == 0 )
		throw string("Divide by zero exception");
	return num1 / num2;
}
int main( void )
{
	try
	{
		int result = ::calculate(10,0);
	}
	catch( const char* ex )
	{
		cout<<"const char*"<<endl;
	}
	catch( string& ex )
	{
		cout<<"string"<<endl;
	}
	catch(...)
	{
		cout<<"Divide by zero exception"<<endl;
	}
}
```

Answers

1. const char*
2. string
3. Divide by zero exception
4. **Abnormal termination**

14. Which of the following line will give a compiler error?

```
class Base
{       };
class Derived : virtual public Base    //Line 1
{       };
int main( void )
{
        Base* ptrBase = new Base;   //Line 2
        Derived* ptrDerived1 = dynamic_cast<Derived*>( ptrBase );       //Line 3
        Derived* ptrDerived2 = reinterpret_cast<Derived*>( ptrBase);     //Line 4
        return 0;
}
```

Answers
1. Line 1
2. Line 2
3. **Line 3**
4. Line 4

15. To find out the true type of object, If we use the null pointer with typeid then output is ____
Answers
1. NullPointer exception
2. bad_typeid exception
3. bad_cast exception
4. No action will be performed.

16. Which stream class is used to perform read as well as write operations on file?
Answers
1. iostream
2. fstream
3. ifstream
4. ofstream

17. Which one of the following STL container store elements in adjacent memory locations?
Answers
1. std::vector
2. std::list
3. std::map
4. std::set

18. Which one of the following is not a fundamental datatype in C++?
Answers
1. bool
2. wchar_t
3. char
4. string

19. We can convert pointer of child type into pointer of parent type. It is called _____?
Answers
1. downcasting
2. upcasting
3. narrowing

4. widening

20. What is the output of the following code?

```
class A
{
private:
	class B
	{
	private:
		int number;
	public:
		B( void ) : number( 10 ){	}
		friend class A;
	};
public:
	class C
	{
	public:
		void print( void )
		{
			B obj;
			cout<<"Number	:	"<<obj.number<<endl;
		}
	};
};
int main( void )
{
	A::C obj;
	obj.print();
	return 0;
}
```

Answers
1. We can not write class inside class in C++.
2. Error : Class C is not a friend of class B
3. Error : Class B is not a friend of class C
4. **Number : 10**

21. by default all data members of class are _____ and all data members of struct are____.
Answers
1. public , private
2. private , public
3. public , public
4. private , private

22. State of object can be modified in ----------?
Answers
1. all non constant member functions of class
2. inspectors member function of class
3. mutator member function of class
4. None of the Above

23. Which of the following way is correct to access static data member with class name?
Answers
1. className->staticDataMember;
2. className.staticDataMember;
3. *(className.staticDataMember);
4. className::staticDataMember;

24. If you want to modify data member inside a constant member function, the data member should be declared as --------.
Answers
1. mutable
2. constant
3. static
4. virtual

25. In c plus plus programming language we can initialized pointer to _____.
In c plus plus programming language we can not initialized reference to ____.
Answers
1. NULL, NULL
2. 0, 0
3. both A and B
4. none of above

26. Which of the following is/are valid ways to allocate memory for an integer by dynamic memory allocation in c plus plus?
Answers
1. int *p = new int(1);
2. int *p= new int; *p = 1;
3. int *p = NULL; p = new int; *p=1;
4. **All of the above**

27. Which of the following is true statement about new in cpp ?
Answers
1. it is aware of constructor.
2. it is an operator.
3. need to specify number of objects to allocate memory.
4. all of these

28.
```
#include<iostream>
using namespace std;
int main(void)
{
        enum colors{ RED,BLUE=-1,GREEN,YELLOW=-1 };
        cout<<YELLOW<<" "<<GREEN<<" "<<BLUE<<" "<<RED<<endl;
        return 0;
}
```
Answers
1. Compile time Error
2. 0 , -1 , -2 , -3 , -1
3. 0 , -1 , 0 , -1

4. **-1 , 0 , -1 , 0**

29.
```
#include<iostream>
using namespace std;
class democlass
{
   char ch;
        public:
   democlass(char x){
        this->ch = ch;
        }
   democlass(const democlass p) {
        this->ch = p.ch;
   }
   char getch() { return ch; }
};
int main()
{
        democlass objInstance1('A');
        democlass objInstance2 = objInstance2;
        cout << objInstance1.getch();
        return 0;
}
```
Answers
1. A
2. **Compiler time error**
3. Garbage value
4. Run time error

30.
```
#include<iostream>
using namespace std;
class democlass
{
        public:
                democlass()
                {
                cout << "Constructor called "<<endl;
                }
                democlass(const democlass &t)
        {
                cout << "Copy constructor called"<<endl;
        }
};
int main()
{
        democlass *t1=NULL, *t2=NULL;
        t1 = new democlass();
        t2 = new democlass(*t1);
        democlass t3 = *t1;
        democlass t4;
        t4 = t3;
```

```
    return 0;
}
```
Answers
1. Constructor called Constructor called Constructor called Copy Constructor called Copy Constructor called Constructor called Copy Constructor called
2. Constructor called Copy Constructor called Copy Constructor called Constructor called Copy Constructor called
3. **Constructor called Copy Constructor called Copy Constructor called Constructor called**
4. Constructor called Constructor called Constructor called Copy Constructor cal

32. in diamond problem if we consider class A,B,C,D then constructor calling sequence for following code.
```
class A {};
class B: virtual public A {};
class C: virtual public A {};
class D: public C, public B {};
```
Answers
1. class A , class C , class A , class B , class D
2. class A , class B , class A , class C , class D
3. class A , class B , class C , class D
4. class A , class C , class B , class D

33. function overloading is --------- and function overriding is --------.
Answers
1. Static polymorphism , Dynamic polymorphism
2. Static binding , dynamic binding
3. early binding , late binding
4. **All of above**

34. to make member function Constant in C++ which the correct way of following
Answers
1. void functionname()
2. const void functionname()
3. void functionname() const
4. void const functionname()

35. if you want to do type conversion between incompatible types then we should use ------- operator.
Answers
1. reinterpret_cast
2. static_cast
3. dynamic_cast
4. const_cast

36. _____ can occur with in same class or same scope(in global functions) and _____ occurs when one class is inherited from another class.
Answers
1. Function overriding,Function overloading
2. **Function overloading,Function overriding**

3. Virtual Function,Pure Virtual Function
4. None of the above

37. A _____ is a member function that is declared within a base class and it is not compulsory to redefined by a derived class. and _____ is a member function that is declared within a base class and it is compulsory to redefined by a derived class.
Answers
1. pure virtual Function, virtual Function
2. constant member function, virtual member function
3. virtual Function, pure virtual Function
4. static member function, virtual member function

38. when object of any user defined class having dynamic memory allocation in constructor of class is thrown by exception handling. you should throw it by ____ as argument to avoid extra functions calls.
Answers
1. value.
2. reference
3. both of above
4. none of above

39. Which of the following are the default standard streams in C++.

1. cin
2 cout
3 cerr
4 clog
Answers
1. 1,2
2. 1,2,3
3. 1,2,4
4. **1,2,3,4**

40. To delete all content of file while opening it, which mode is use _____.
Answers
1. ios::trunc
2. ios::truncate
3. ios::app
4. ios::write