

cpp mcq exam

Total points 20/26 ?


The respondent's email (**sarthaksalgar7@gmail.com**) was recorded on submission of this form.

✓ What is the output of the program?

1/1

```
#include<iostream>
using namespace std;
class Base
{
    public:
    Base()
    {
        cout<< "Base OK. ";
    }
    ~Base()
    {
        cout<< "Base OK. ";
    }
};
class Derived: public Base
{
    public:
    Derived()
    {
        cout<< "Derived OK. ";
    }
    ~Derived()
    {
        cout<< "Derived OK. ";
    }
};
int main()
{
    Derived b;
    return 0;
}
```

- ☐ Base OK. Derived OK.
- ☐ Base OK. Base OK.Derived OK. Derived OK.
- ☐ Base OK. Derived OK.Base OK. Derived OK.

 Base OK. Derived OK. Derived OK. Base OK



✗ Which of the following offers a programmer the facility of using a specific class object into other classes? 0/1

- ☐ Polymorphism
- ☐ Abstraction
- ☒ Inheritance
- ☐ Composition



Correct answer

- ☒ Composition

✓ Which of the following type of class allows only one object of it to be created? 1/1

- ☐ Virtual class
- ☐ Abstract class
- ☒ Singleton class
- ☐ Friend class



✓ Find the output of the following program.

1/1

```
int main()
{
    char ch[] = "c++ programs";
    int i = sizeof(ch);
    cout << i << endl;
}
```

☒ 13



☐ 12

☐ 1

☐ 4

✓ Which of the following definition best describes the concept of polymorphism?

1/1

☐ It is the ability to process the many messages and data in one way

☐ It is the ability to process the undefined messages or data in at least one way

☒ It is the ability to process the message or data in more than one form



☐ It is the ability to process the message or data in only one form

✓ What is the output of the Program?

1/1

```
class Base
{
    int x, y;
public:
    int z;
public:
    Base()
    {
        x = y = z = 0;
    }
    void Display(void)
    {
        cout<< x << " " << y << " " << z << endl;
    }
};

class Derived : public Base
{
    int x, y;
public:
    Derived(int xx = 65, int yy = 66)
    {
        y = xx;
        x = yy;
    }
    void Display(void)
    {
        cout<< x << " " << y << " "<<z;

    }
};

int main()
{
    Derived objD;
    objD.Display();
    return 0;
}
```

- ☒ 66 65 0
- ☐ 66 65 garbage
- ☐ none of the above.



NAME

Sarthak Salgar, Sarvesh Kulkarni

✗ Which one of the following given statements is not true about the references in C++?

0/1

- ☒ Array of reference cannot be created. ✗
- ☐ A reference cannot refer to a constant value
- ☐ A reference cannot be NULL
- ☐ Once a reference is created, it cannot be later made to reference another object; it cannot be reset

Correct answer

- ☒ A reference cannot refer to a constant value

✓ Which of the following statements is correct about the friend function in C++ programming language? 1/1

- ☐ A friend function can access the private members of a class
- ☐ A friend function is able to access protected members of a class
- ☐ A friend function is able to access the public members of a class
- ☒ All of the above



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

```
#include <iostream>
using namespace std;
int main ()
{
    char str1[10] = "Hello";
    char str2[10] = "World";
    char str3[10];
    int len ;
    strcpy( str3, str1);
    strcat( str1, str2);
    len = strlen(str1);
    cout<<len<<endl;
    return 0;
}
```

- ☐ 5
- ☐ 12
- ☒ 10
- ☐ 11



✗ Which of the following type of data member can be shared by all instances of its class? 0/1

- ☒ Public
- ☐ protected
- ☐ Static
- ☐ Friend

✗

Correct answer

- ☒ Static

✓ Which one of the following statements correctly refers to the Delete and Delete[] in C++ programming language? 1/1

- ☐ The "delete" is used for deleting the standard objects, while on the other hand, the "Delete[]" is used to delete the pointer objects
- ☐ The "Delete" is a type of keyword, whereas the "Delete[]" is a type of identifier
- ☒ The "Delete" is used for deleting a single standard object, whereas the "Delete[]" is used for deleting an array of the multiple objects ✓
- ☐ Delete is syntactically correct although, if the Delete[] is used, it will obtain an error

✓ `#include <iostream>`
`using namespace std;`
`int main()`
`{`
`int array[] = {10, 20, 30};`
`cout << -2[array];`
`return 0;`
`}`

1/1

- ☒ -30
- ☐ compiler error
- ☐ garbage value
- ☐ -15



✗ Which of the following statement is correct?

0/1

- ☐ Overloaded functions can accept same number of arguments.
- ☐ Overloaded functions always return value of same data type.
- ☐ Overloaded functions can accept only same number and same type of arguments.
- ☒ Overloaded functions can accept only different number and different type of arguments.



Correct answer

- ☒ Overloaded functions can accept same number of arguments.

- ✓ Which of the following statement is true about the new and malloc? 1/1
- I. The "new" is a type of operator while "malloc" is a kind of function
II. "new" invokes a constructor, whereas "malloc" does not invoke the constructor
III. "malloc" returns void pointer and also needed to typecast whereas "new" returns required the pointer
- ☐ Only I
- ☐ Both I and II
- ☒ I, II, III ✓
- ☐ None of the above

- ✓ Which one of the following is the correct way to declare a pure virtual function? 1/1
- ☐ virtual void Display(void){0};
- ☐ virtual void Display = 0;
- ☒ virtual void Display(void) = 0; ✓
- ☐ void Display(void) = 0;

✗ Which of the following can be considered as the correct syntax for declaring an array of pointers of integers that has a size of 10 in C++?

0/1

- ☒ `int *arr = new int*[10]`
- ☐ `int *arr = new int[10];`
- ☐ `int arr = new int[10];`
- ☐ `int **arr = new int*[10];`

✗

Correct answer

- ☒ `int **arr = new int*[10];`

✗ Which of the following statement is not true about C++?

0/1

- ☒ A class cannot have the data members as pointer.
- ☐ Dynamic objects in c++ can be created only with the help of new operator.
- ☐ Using abstract class we can achieve runtime polymorphism.
- ☐ Static function can be overridden.

✗

Correct answer

- ☒ Static function can be overridden.

✓ Which of the following statement is correct about Virtual Inheritance? 1/1

- ☐ It is a technique to ensure that a private member of a base class can be accessed
- ☐ It is a technique to optimize the multiple inheritances
- ☐ It is a technique to avoid the multiple inheritances of the classes
- ☒ It is a C++ technique to avoid multiple copies of the base class into the derived or child classes ✓

✓ What is the output of the Program?

1/1

```
#include <iostream>
using namespace std;
class Demo
{
    int x, y;
public:
    void SetValue(int &a, int &b)
    {
        a = 100;
        x = a;
        y = b;
        Display();
    }
    void Display()
    {
        cout<< x << " " << y;
    }
};
int main()
{
    int x = 10;
    Demo d;
    d.SetValue(x, x);
    return 0;
}
```

- ☐ The program will print the output 100 10.
- ☒ The program will print the output 100 100. ✓
- ☐ The program will print the output 100 garbage.
- ☐ It will result in a compile time error.

PRN

240845920077, 240845920078
.....

✓ Which of the following statement will be correct if the function has three arguments passed to it? 1/1

- ☒ The trailing argument will be the default argument. ✓
- ☐ The first argument will be the default argument.
- ☐ The middle argument will be the default argument.
- ☐ All the argument will be the default argument.

✓ Among the following, which statement is correct about the Modularity? 1/1

- ☐ Modularity means hiding the parts of the program
- ☒ Modularity refers to dividing a program into subsequent small modules or independent parts ✓
- ☐ It refers to overloading the program's part
- ☐ Modularity refers to wrapping the data and its functionality into a single entity

✓ Which one of the following cannot be used with the virtual keyword? 1/1

- ☐ Destructor
- ☐ Member function
- ☒ Constructor
- ☐ None of the above



✓ Which of the following functions must use the reference in the argument list to avoid chain of calls? 1/1

- ☒ Copy constructor
- ☐ Virtual Function
- ☐ Friend Function
- ☐ Operator Function



✓ What is the output of the Program?

1/1

```
#include <iostream>
using namespace std;
class Demo
{
    public:
    Demo(int xx)
    {
        cout<< xx;
    }
    ~Demo()
    {
        cout<< "Final";
    }
};
int main()
{
    Demo *ptr = new Demo('B');
    return 0;
}
```

- ☐ Compile time error
- ☐ B
- ☐ garbage
- ☒ 66



✓ Which of the following is used for implementing the late binding? 1/1

- ☐ Operator Functions
- ☒ Virtual Functions
- ☐ new Operator
- ☐ Static Functions



✓ Assume that Honda is an instance of the Car class, and that Car class has a member function named run. Which of the following is a correct call to the run function? 1/1

- ☐ Honda->run;
- ☒ Honda.run();
- ☐ run()
- ☐ Honda()



This content is neither created nor endorsed by Google. [Report Abuse](#) - [Terms of Service](#) - [Privacy Policy](#)

Google Forms

