Infosys Springboard | Explore | EN | What do you want to learn?

Programming Using C++

C++ is an object-oriented programming language that was first developed by Bjarne Stroustrup. It was originally created to be an extension of the C language, which i...More

Start

Learning Progress

Status: Completed

Completion Certificate

You can earn a certificate on completing this content

Overview | Contents | Discussions

What you will learn

To introduce the implementation issues of Basic Object Oriented Concepts in C++. To enable the participants to write application programs using the additional object oriented features of C++. To introduce advanced features of C++ namely Templates, STL , Namespaces, Exception Handling.

Skills you will gain

C and C++

Authors/Creators

MR Malarvizhi Rathinam01

BV Balasundari Vaidyanathan

Show Curators/Contacts

At a glance

- Course
- 24h 18m
- Beginner Level
- Free
- Infosys Wingspan
- EN
- C++, basics, cpp, programming, using, Basics

Thank you. Your test submitted.

You have cleared this assessment.

Obtained Percentage	Obtained Marks
65 %	13 / 20

Best Attempt Score:65 % on 23-03-2025

What is the output for the following program?

```
int main(){
    int a[5] = {5, 1, 15, 20, 25};
    int i, j, m;
    i = ++a[1];
    j = a[1]++;
    m = a[i++];
    cout << i << " " << j << " " << m;
}
```

- (●) 3, 2, 15
- ( ) 2, 3, 20
- ( ) 2, 1, 15
- ( ) 1, 2, 20

Find the output for the below code snippet?

```
enum demo {
   A = 10, B, C
};
int main() {
   cout << A << B<< C;
   return 0;
}
```

- (●) 101112
- ( ) 101010
- ( ) 1098
- ( ) 10119

Consider the following recursive function fun(x, y). What is the value of fun(4, 3)

```
int fun(int x, int y)  {
   if (x == 0)
      return y;
   return fun(x - 1,  x + y);
}
```

- (●) 13
- ( ) 12
- ( ) 9
- ( ) 10

Warning

This operation is disab

Ok

Predict the output of the following code:-

```
int main() {
    unsigned int m = 32;
    cout << (m >> 1) << " " << ~m;
    return 0;
}
```

- ○ 16 5
- ◉ 64 -32
- ○ 16 33
- ○ 16 -33

Procedural programming is also referred to as _____.

- (●) Imperative programming
- ( ) Declarative programming
- ( ) Routine programming
- ( ) Functional programming
- ( ) Object-oriented programming

What is the output for the below code:

```
class BaseA {
   public:
      BaseA() {
         cout << " Base A's constructor called" << endl;
      }
};

class BaseB {
   public:
      BaseB() {
         cout << "Base B's constructor called" << endl;
      }
};

class Derived : public BaseA, public BaseB {
   public:
      Derived() {
         cout << "Derived's constructor called" << endl;
      }
};

int main() {
   Derived derived;
   return 0;
}
```


- ◯ Compiler Dependent
- ◯ Base A's constructor called Base B's constructor called Derived's constructor called
- ◯ Base B's constructor called Base A's constructor called Derived's constructor called
- ◉ Compiler Error

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

```
class base {
   public:
   void fun_1() { cout << "base-1"; }
   virtual void fun_2() { cout << "base-2"; }
   virtual void fun_3() { cout << "base-3"; }
   virtual void fun_4() { cout << "base-4"; }
};

class derived : public base {
   public:
      void fun_1() { cout << "derived-1"; }
      void fun_2() { cout << "derived-2"; }
      void fun_4(int x) { cout << "derived-4"; }
};

int main() {
   base* p;
   derived obj1;
   p = &obj1;
   p->fun_1();
   p->fun_2();
   p->fun_3();
   p->fun_4();
}
```

- ○ Compile Time Error
- ○ base-1 derived-2 base-3 base-4
- ◉ base-1 derived-2 derived-3 derived-4
- ○ base-1 derived-2 base-3 derived-4

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

```
template <typename T>
T max(T x, T y) {
    return (x > y)? x : y;
}
int main() {
    cout << max(3.5) << std::endl;
    cout << max(3.0,5.0) << std::endl;
    cout << max(3,5.0) << std::endl;
    return 0;
}
```

Warning

This operation is dis

- (●) Compiler Error in all cout statements as data
- ( ) 5 5.0 5.0
- ( ) Compiler Error in last cout statement as call to max is ambiguous
- ( ) None of the above

What will be the output for the below code snippet:

```
void square (int *x) {
    *x = (*x + 1) * (*x);
}

int main ( ) {
    int num = 10;
    square(&num);
    cout << num;
    return 0;
}
```

- ○ 100
- ○ Compile Time Error
- ○ 144
- ◉ 110

Find the Output for the below code snippet:

```
int main() {
    int i;
    int arr[5] = {1};
    for (i = 0; i < 5; i++)
        cout << arr[i] << " ";
    return 0;
}
```

- ○ 1 followed by four garbage values
- ◉ 1 0 0 0 0
- ○ 1 1 1 1 1
- ○ 0 0 0 0 0

Predict the output the of following program.

```
class B;

class A {
      int a;
   public:
      A():a(0) { }
      void show(A& x, B& y);
};

class B {
   private:
      int b;
   public:
      B():b(0) { }
      friend void A::show(A& x, B& y);
};

void A::show(A& x, B& y) {
   x.a = 10;
   cout << "A::a=" << x.a << " B::b=" << y.b;
}

int main() {
   A a;
   B b;
   a.show(a,b);
   return 0;
}
```

- ◯ Compiler Error
- ◉ A::a=10 B::b=0
- ◯ A::a=0 B::b=0
- ◯ None of the above

What is the output of the following C++ code?

```
int fun(int x = 0, int y = 0, int z) {
    return (x + y + z);
}

int main() {
    cout << fun(10);
    return 0;
}
```

- ○ 10
- ○ 0
- ○ 20
- ◉ Compilation Error

What is the output for the below code snippet:

```
void show(int* arr) {
   int n = sizeof(arr)/sizeof(arr[0]);
   int i;
   for (i = 0; i < n; i++)
      cout<< arr[i]<< " " ;
}

int main() {
   int arr[] = {1, 2, 3, 4, 5, 6, 7, 8};
   show(arr);
   return 0;
}
```

- ○ 1 2 3 4 5 6 7 8
- ○ Compilation error
- ○ 1 2
- ◉ Runtime error

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

```
int main(int argc, char const *argv[]) {
    char str[] = "Hola Infoscians";
    cout<<str[0];
    return 0;
}
```

- ○ H
- ○ Hola
- ◉ Error
- ○ 0

Choose the correct statement for the below program

```
int main() {
   int *pInt;
   int **ppInt1;
   int **ppInt2;

   pInt = (int*)malloc(sizeof(int));
   ppInt1 = (int**)malloc(10*sizeof(int*));
   ppInt2 = (int**)malloc(10*sizeof(int*));

   free(pInt);
   free(ppInt1);
   free(*ppInt2);
   return 0;
}
```


- ◯ malloc() for ppInt1 and ppInt2 isn't correct. It'll give compile time error
- ◯ free(*ppInt2) is not correct. It'll give compile time error
- ◉ free(*ppInt2) is not correct. It'll give run time error
- ◯ No issue with any of the malloc() and free() i.e. no compile/run time error

Which of the following is true about the following program

```
class Test {
   public:
      int i;
      void get();
};
void Test::get() {
   std::cout << "Enter the value of i: ";
   std::cin >> i;
}
Test t;  // Global object
int main() {
   Test t;  // local object
   t.get();
   std::cout << "value of i in local t: "<<t.i<<'n';
   ::t.get();
   std::cout << "value of i in global t: "<<::t.i<<'n';
   return 0;
}
```

Warning

This operation

- ◯ Compiler Error: Cannot have two objects with same class name
- ◯ Compiler Error in Line "::t.get();"
- ◉ Compiles and runs fine
- ◯ None of the above

What will be the output for the below code snippet?

```
int main() {
   try {
      throw 100;
   }
   catch (...) {
      cout << "default exception";
   }
   catch (int param) {
      cout << "int exception";
   }
   return 0;
}
```

- (●) Compile time error
- ( ) Run time error
- ( ) Default exception
- ( ) int exception

What is the output of this program?

```
int main() {
   struct employee {
      int empid;
      char empname[25];
   };
   employee e;
   e.empid = 1001;
   strcpy(e.empname, "Peter");
   cout << e.empid << endl;
   cout << e.empname << endl;
   return 0;
}
```

- (●) 1001 Peter
- ( ) Peter Peter
- ( ) Compile time error
- ( ) None of the above

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

```
class Base {
   public:
      virtual void print() const = 0;
};
class DerivedOne : virtual public Base {
   public:
   void print() const {
      cout << "1";
   }
};
class DerivedTwo : virtual public Base {
   public:
      void print() const {
         cout << "2";
      }
};
class Multiple : public DerivedOne, DerivedTwo {
   public:
      void print() const {
         DerivedTwo::print();
      }
};
int main() {
   Multiple both;
   DerivedOne one;
   DerivedTwo two;
   Base *array[ 3 ];
   array[ 0 ] = &both;
   array[ 1 ] = &one;
   array[ 2 ] = &two;
   for ( int i = 0; i < 3; i++ )
      array[ i ] -> print();
   return 0;
}
```

- ○ 121
- ○ 12
- ◉ 212
- ○ 215

What is the output of the following code?

```
class Test {
   private:
      static int count;
   public:
      Test& fun();
};

int Test::count = 0;

Test& Test::fun() {
   Test::count++;
   cout << Test::count << " ";
   return *this;
}

int main(){
   Test t;
   t.fun().fun().fun().fun();
   return 0;
}
```

- ◯ Compiler Error
- ◯ 4 4 4 4
- ◯ 1 1 1 1
- ◉ 1 2 3 4