

X


<https://swayam.gov.in>

https://swayam.gov.in/nc_details/NPTEL

reviewer4@nptel.iitm.ac.in ▾

[NPTEL \(https://swayam.gov.in/explorer?ncCode=NPTEL\)](https://swayam.gov.in/explorer?ncCode=NPTEL) » [Programming in C++ \(course\)](#)
[Announcements \(announcements\)](#) [About the Course \(https://swayam.gov.in/nd1_noc20_cs07/preview\)](#)
[Ask a Question \(forum\)](#) [Progress \(student/home\)](#) [Mentor \(student/mentor\)](#)

Unit 8 - Week 6

Course outline

How does an NPTEL online course work?

Week 0

Week 1

Week 2

Week 3

Week 4

Week 5

Week 6

- Module 26 :
Dynamic Binding : Part I
(Lecture 41)
(unit?
unit=8&lesson=58)
- Module 27 :
Dynamic Binding
(Polymorphism)
: Part II (Lecture 42)
(unit?
unit=8&lesson=59)

Assignment 6

The due date for submitting this assignment has passed. **Due on 2020-03-11, 23:59 IST.**
As per our records you have not submitted this assignment.

- 1) Consider the following program. 2 points

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

class A {
    int a;
public:
    A(int x) : a(x) { }
    void print() { cout << a << endl; }
};

int main() {
    A *a1 = new A(5);

    void *x;
    x = new int(10);
    x = a1;

    x->print();

    return 0;
}
```

What will be the output/error of the above code?

- a) 5

Module 28 :
Dynamic
Binding
(Polymorphism)
: Part III
(Lecture 43)
(unit?
unit=8&lesson=60)

Module 29 :
Dynamic
Binding
(Polymorphism)
Part IV (Lecture
44) (unit?
unit=8&lesson=61)

Module 30 :
Dynamic
Binding
(Polymorphism)
: Part V (Lecture
45) (unit?
unit=8&lesson=62)

Lecture
Materials (unit?
unit=8&lesson=63)

Quiz :
Assignment 6
(assessment?
name=134)

W6_ProgrammingQs-
1
(/noc20_cs07/progassignment?
name=136)

W6_ProgrammingQs-
2
(/noc20_cs07/progassignment?
name=137)

W6_ProgrammingQs-
3
(/noc20_cs07/progassignment?
name=138)

W6_ProgrammingQs-
4
(/noc20_cs07/progassignment?
name=139)

Feedback For
Week 6 (unit?
unit=8&lesson=146)

Week 7

Week 8

**DOWNLOAD
VIDEOS**

- b) 10
 c) Error: Request for member 'print' in 'x', which is of non-class type 'void*'
 d) Error: Cannot convert 'A' to 'void*' in assignment

No, the answer is incorrect.

Score: 0

Accepted Answers:

c) Error: Request for member 'print' in 'x', which is of non-class type 'void*'

2)

Consider the following program.

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

class A {
public:
    void f() { }
};

class B : public A {
public:
    ----- // LINE-1

    void f(int a) {}
};

int main() {
    B b;
    b.f(); // LINE-2

    return 0;
}
```

Fill in the blanks in LINE-1 so that the function call at LINE-2 will not give any error.

- a) using A::f;
 b) int f(int a) {}
 c) void f() {}
 d) int f() {}

No, the answer is incorrect.

Score: 0

Accepted Answers:

a) using A::f;
c) void f() {}
d) int f() {}

0 points

Text Transcripts

Assignment
SolutionLive Interactive
Session

Books

3) Consider the following program.

2 points

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

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

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

int main() {
    A *a = new A();
    A *b = new B();

    delete a;
    delete b;

    return 0;
}
```

What will be the output of the following code?

- a) A()
A()
B()
~A()
~B()
~A()
- b) A()
A()
B()
~A()
~A()
~B()
- c) A()
A()
B()
~A()
~A()
- d) A()
A()
B()
~B()
~B()

No, the answer is incorrect.

Score: 0

Accepted Answers:

a) A()
 A()
 B()
 ~A()
 ~B()
 ~A()

4) Consider the program below.

2 points

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

class A {
public:
    virtual void f1() { cout << "A::f1" << endl; }
    void f2() { cout << "A::f2" << endl; }
};

class B : public A {
public:
    void f1() { cout << "B::f1" << endl; }
    virtual void f2() { cout << "B::f2" << endl; }
};

class C : public B {
public:
    void f1() { cout << "C::f1" << endl; }
    void f2() { cout << "C::f2" << endl; }
};

int main() {
    B *b = new C();
    b->f1();
    b->f2();

    return 0;
}
```

What will be the output of the above code.

- a) A::f1
 B::f2
- b) B::f1
 C::f2
- c) C::f1
 C::f2
- d) B::f1
 B::f2

No, the answer is incorrect.

Score: 0

Accepted Answers:

c) C::f1
 C::f2

2 points

5) Consider the program below.

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

class A { public: int a = 5; };
class B { public: int b = 8; };
class C { public: double c = 5.8; };

int main() {
    A t1;
    A *p = &t1;
    B *q = (B*)p;
    C *r = (C*)p;

    cout << p->a << endl;
    cout << q->b << endl;
    cout << r->c << endl;

    return 0;
}
```

What will be output / error?

- a) 5
5
<garbage-value>
- b) 5
8
<garbage-value>
- c) 5
8
5.8
- d) 5
<garbage-value>
<garbage-value>

No, the answer is incorrect.

Score: 0

Accepted Answers:

a) 5
5
<garbage-value>

2 points

6) Consider the following program.

```

#include <iostream>
using namespace std;

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

class Derived : public Base {
public:
    void f() {
        ----- // LINE-1

        cout << "Derived" << endl;
    }
};

void Base::f() {
    cout << "Base" << endl;
}

int main() {
    Base *b = new Derived();
    b->f();

    return 0;
}

```

Choose the correct missing statement/s at LINE-1 such that it will print

```

Base
Derived

```

- a) Base::f();
- b) Base.f();
- c) using Base::f();
- d) Base b; b.f();

No, the answer is incorrect.

Score: 0

Accepted Answers:

a) Base::f();

7) Consider the program below.

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

class Base {
public:
    void f() { cout << "Base" << endl; }
};

class Derived1 : public Base {
public:
    void f() { cout << "Derived1" << endl; }
};

class Derived2 : public Base {
public:
    void f() { cout << "Derived2" << endl; }
};

class ReDerived : public Derived1, public Derived2 {
public:
    using Base::f;
};

int main() {
    ReDerived *b = new ReDerived();
    b->f();

    return 0;
}
```

What will be the output/error we will get?

- a) Base
- b) Derived1
Derived2
- c) Compile error: 'Base' is an ambiguous base of 'ReDerived'.
- d) Compile error: type 'Base' is not a base type for type 'ReDerived'.

No, the answer is incorrect.

Score: 0

Accepted Answers:

c) Compile error: 'Base' is an ambiguous base of 'ReDerived'.

8) Consider the following program.

2 points

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

class cls1 {
public:
    virtual void f() = 0;
};

class cls2 {
public:
    void g() {
        cout << "cls2" << endl;
    }
};

int main() {
    cout << sizeof(cls1) << " " << sizeof(cls2) << endl;

    return 0;
}
```

What will be the output? (if `sizeof(void*) = 8`)

- a) 8 0
- b) 8 1
- c) 1 1
- d) 8 8

No, the answer is incorrect.

Score: 0

Accepted Answers:

b) 8 1

2 points

9) Consider the following program.

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

class A {
    int x = 0;
public:
    void getVal() { cout << x + 5 << endl; }
};

class B : public A {
    int x = 5;
public:
    void getVal() { cout << x + 5 << endl; }
};

int main() {
    A *a = new B();
    a->A::getVal();

    return 0;
}
```

What will be the output/error?

- a) 5
- b) 10
- c) Compilation Error: cannot be called A::getVal using pointer of class A.
- d) 0

No, the answer is incorrect.

Score: 0

Accepted Answers:

a) 5

10 What is the output of the following program?

2 points

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

class Base {
public:
    virtual void f() { cout << "Base::f()\n"; }
    void g() { cout << "Base::g()\n"; }
};

class Derived : public Base {
public:
    void f() { cout << "Derived::f()\n"; }
    virtual void g() { cout << "Derived::g()\n"; }
};

int main() {
    Base b;
    Derived d;

    b = d;          // LINE-1
    b.f();
    b.g();

    Base& rb = d; // LINE-2
    rb.f();
    rb.g();

    return 0;
}
```

- a) Derived::f()
Derived::g()
Derived::f()
Base::g()
- b) Base::f()
Base::g()
Derived::f()
Derived::g()
- c) Derived::f()
Derived::g()
Derived::f()
Derived::g()
- d) Base::f()
Base::g()
Derived::f()
Base::g()

No, the answer is incorrect.

Score: 0

Accepted Answers:

d) Base::f()

Base::g()

Derived::f()

Base::g()

