X

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

sahilrao736@gmail.com

NPTEL (https://swayam.gov.in/explorer?ncCode=NPTEL) » **Programming in C++ (course)**

Announcements (announcements) **About the Course (preview)** Ask a Question (forum)

Progress (student/home) Mentor (student/mentor)

Unit 9 - Week 7

Course outline

How does an NPTEL online course work?

Week 0

Week 1

Week 2

Week 3

Week 4

Week 5

Week 6

Week 7

- Module 31 : Virtual Function Table (Lecture 46) (unit?unit=80&lesson=81)
- Module 32 : Type Casting and Cast Operators : Part I (Lecture 47)

Assignment 7

The due date for submitting this assignment has passed. **Due on 2020-11-04, 23:59 IST.**

Assignment submitted on 2020-11-04, 14:45 IST

1) ***2 points***

(unit?
unit=80&lesson=82)

- Module 33 : Type Casting and Cast Operators : Part II (Lecture 48) (unit? unit=80&lesson=83)
- Module 34 : Type Casting and Cast Operators : Part III (Lecture 49) (unit? unit=80&lesson=84)
- Module 35 : Multiple Inheritance (Lecture 50) (unit? unit=80&lesson=85)
- Module 35 : Multiple Inheritance (Contd.) (Lecture 51) (unit? unit=80&lesson=86)
- Lecture Materials (unit? unit=80&lesson=87)
- **Quiz : Assignment 7 (assessment? name=170)**
- W7_Programming-Qs1 (/noc20_cs57/progassignment? name=171)
- W7_Programming-Qs2 (/noc20_cs57/progassignment? name=172)
- W7_Programming-Qs3 (/noc20_cs57/progassignment? name=173)
- W7_Programming-Qs4 (/noc20_cs57/progassignment? name=174)
- ○ Feedback For Week 7 (unit? unit=80&lesson=88)

Week 8

Consider the program below.

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

class employee {
    int emp_id;
    string name;
public:
    employee(int _emp_id, string _name) : emp_id(_emp_id), name(_name) { }

    void update(int id, string na) const {
        (____________________)->emp_id = id;   // LINE-1
        (____________________)->name = na;     // LINE-2
    }

    void showInfo() const {
        cout << emp_id << " : " << name;
    }
};

int main() {
    const employee e(3, "Raj");

    e.update(30, "Rajan");
    e.showInfo();

    return 0;
}
```

Fill in the blank at LINE-1 and LINE-2 such that the output is: `30 : Rajan`

☑ a) const_cast <employee*> (this)

□ b) static_cast <employee*> (this)

□ c) dynamic_cast <employee*> (this)

☑ d) ((employee*) (this))

Yes, the answer is correct.
Score: 2

Accepted Answers:

a) const_cast <employee*> (this)

d) ((employee*) (this))

2) *2 points*

DOWNLOAD VIDEOS

Text Transcripts

Assignment Solution

Books

Live Interactive Session

Programming Test (11th Dec): Session-1 (10.00AM - 11.00AM)

Programming Test (11th Dec): Session-2 (8.00PM - 9.00PM)

Consider the program below.

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

class Base {
    int data;
public:
    Base() { cout << "Base()" << " "; }
    Base(int _x) { cout << "Base(x)" << " "; }
};

class Derived1 : virtual public Base {
public:
    Derived1() { cout << "Derived1()" << " "; }
    Derived1(int _x) : Base(_x) { cout << "Derived1(x)" << " "; }
};

class Derived2 : virtual public Base {
public:
    Derived2() { cout << "Derived2()" << " "; }
    Derived2(int _x) : Base(_x) { cout << "Derived2(x)" << " "; }
};

class ReDerived : public Derived1, public Derived2 {
public:
    ReDerived() { cout << "ReDerived()" << " "; }
    ReDerived(int x) : Derived1(x), Derived2(x) { cout << "ReDerived(x)" << " "; }
};
   int main() {
       ReDerived(5);

       return 0;
   }
```

How many virtual table will be set up by the compiler?

○ a) Base(x) Derived1(x) Base(x) Derived2(x) ReDerived(x)

○ b) Base() Derived1(x) Base() Derived2(x) ReDerived(x)

● c) Base() Derived1(x) Derived2(x) ReDerived(x)

○ d) Base() Base(x) Derived1(x) Derived2(x) ReDerived(x)

Yes, the answer is correct.
Score: 2

Accepted Answers:
c) Base() Derived1(x) Derived2(x) ReDerived(x)

3) Consider the following program. **_2 points_**

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

int incr(int* ptr) {
    return (*ptr)++;
}

int main() {
    int val = 10;
    const int *ptr = &val;

    val = incr(____________________);    // LINE-1
    cout << val;

    return 0;
}
```

Fill in the blank at LINE-1 so that it will print 10.

● a) `const_cast <int *>(ptr)`

○ b) `static_cast <int *>(ptr)`

○ c) `dynamic_cast <int *>(ptr)`

○ d) `reinterpret_cast <int *>(ptr)`

Yes, the answer is correct.
Score: 2

Accepted Answers:
a) `const_cast <int *>(ptr)`

4) Consider the below class hierarchy. **2 points**

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

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

class B : public A {
public:
    virtual void g() { }
    void h() { }
    virtual void i();
};

class C : public B {
public:
    void g() { }
    virtual void h() { }
};
```

What will be the virtual function table for class C?

○ a)
```
A::f(A* const)
C::g(C* const)
C::h(C* const)
B::i(B* const)
```

● b)
```
A::f(A* const)
B::g(B* const)
C::h(C* const)
B::i(B* const)
```

○ c)
```
A::f(A* const)
B::g(B* const)
B::h(B* const)
C::i(C* const)
```

○ d)
```
A::f(A* const)
B::g(C* const)
C::h(C* const)
C::i(C* const)
```

No, the answer is incorrect.
Score: 0

Accepted Answers:

a)
```
A::f(A* const)
C::g(C* const)
C::h(C* const)
B::i(B* const)
```

5) How many virtual tables will be created for the following program: *2 points*

```
class A { public: virtual void f() { } };
class B : public A { };
class C : public A { public: void g() {} };
class D : public B, public C{ public: void g(){ }};
```

○ a) 1

○ b) 2

○ c) 3

● d) 4

Yes, the answer is correct.
Score: 2

Accepted Answers:
d) 4

6) *2 points*

Which statement in the following program generates compilation error?

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

int main() {
    int i = 10;
    double d = 3.14;
    int *ip = &i;
    double *pd;

    i = static_cast<int>(d);          // statement-1
    d = static_cast<double>(i);       // statement-2
    pd = static_cast<double*>(ip);    // statement-3
    i = static_cast<int>(&i);         // statement-4

    return 0;
}
```

□ a) statement-1

□ b) statement-2

☑ c) statement-3

☑ d) statement-4

Yes, the answer is correct.
Score: 2

Accepted Answers:
c) statement-3
d) statement-4

7) Consider the following code segment. **2 points**

```
struct st1 { };
struct st2 { };
st1* s1 = new st1;
st2* s2 = new st2;
```

Which of the following type casting is permissible?

○ a) `st2 = static_cast<st2*>(s1);`

○ b) `st2 = dynamic_cast<st2*>(s1);`

● c) `st2 = reinterpret_cast<st2*>(s1);`

○ d) `st2 = const_cast<st2*>(s1);`

Yes, the answer is correct.
Score: 2

Accepted Answers:
c) `st2 = reinterpret_cast<st2*>(s1);`

8) Consider the below classes. **2 points**

```
class Base {
    public:
        void f() { }
};

class Derived : public Base {
    public:
        void g() { }
};
```

What is/are the appropriate option/s to perform up-casting where

```
    Base b;
    Derived d;
```

☑ a) `Base *bp = &d;`

☑ b) `Base *bp = (Base*)&d;`

□ c) `Derived *dp = &b;`

□ d) `Derived *dp = (Derived*)&d;`

Yes, the answer is correct.
Score: 2

Accepted Answers:
a) `Base *bp = &d;`

b) `Base *bp = (Base*)&d;`

9) **2 points**

Consider the following program.

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

class Parent { virtual void fun() { } };
class Child : public Parent { void fun() { } };

int main() {
    try {
        Parent *pbd = new Child;
        Parent *pbb = new Parent;

        Child *pd = dynamic_cast<Child*>(pbd); // LINE-1
        if (pd == 0)
            cout << "Null pointer on first type-cast" << endl;

        pd = dynamic_cast<Child*>(pbb); // LINE-2
        if (pd == 0)
            cout << "Null pointer on second type-cast" << endl;

        pd = static_cast<Child*>(pbd); // LINE-3
        if (pd == 0)
            cout << "Null pointer on third type-cast" << endl;
    }
    catch (exception& e) {
        cout << "Exception: " << e.what();
    }

    return 0;
       }
```

What will be the output?

○ a) `Null pointer on first type-cast`

○ b) `Null pointer on third type-cast`

○ c) `Exception:  NULL pointer exception`

● d) `Null pointer on second type-cast`

Yes, the answer is correct.
Score: 2

Accepted Answers:

d) `Null pointer on second type-cast`