# CSCI 140 -- In-Class Exercise 1

| Group Members | Contribution (0 to 10) 0 – no contribution, 10 -- most |
| --- | --- |
| Nero Li | 10 |
| Christian Gonzalez | 10 |
| Tanner Khep | 10 |
| Daniel Kang | 10 |

It is best to share a screen via Zoom and work together. Your group can provide the solution in C++ or Java.

Note taker (responsible to collect information and submit it):

Nero Li

Provide the code for classes A, B, and C using the information below. Do not worry about data and actual code for each operation (just the header and {} if needed).

#include <iostream>

using namespace std;

class A

{

    public:

        A()

        {

            cout << "A:Construct\n";

        }

        void print1()

        {

            cout << "A:print1()\n";

        }

        virtual void print2()

        {

            cout << "A:print2()\n";

        }

        virtual void print3()

        {

            cout << "A:print3()\n";

        }

};

class B : public A

{

    public:

        B()

        {

            cout << "B:Construct\n";

        }

        void print2()

        {

            cout << "B:print2()\n";

        }

        void print3()

        {

            cout << "B:print3()\n";

        }

};

class C : public A

{

    public:

        C()

        {

            cout << "C:Construct\n";

        }

        void print1()

        {

            cout << "C:print1()\n";

        }

        void print2()

        {

            cout << "C:print2()\n";

        }

        void print3()

        {

            cout << "C:print3()\n";

        }

};

int main()

{

    A \*pA;

    pA = new A();   // syntax error

    pA = new B();   // create an object of type B

    pA->print1();   // run A’s print1()

    pA->print2();   // run B’s print2()

    pA->print3();   // run B’s print3()

    pA = new C();   // create an object of type C

    pA->print1();   // run A’s print1()

    pA->print2();   // run C’s print2()

    pA->print3();   // run C’s print3()

    B \*pB;

    pB = new B();   // create an object of type B

    pB->print1();   // run A’s print1()

    pB->print2();   // run B’s print2()

    pB->print3();   // run B’s print3()

    C \*pC;

    pC = new C();   // create an object of type C

    pC->print1();   // run C’s print1()

    pC->print2();   // run C’s print2()

    pC->print3();   // run C’s print3()

    return 0;

}