

# CC3002: Ejercicios usando herencia, polimorfismo y enlace dinamico

Prof: Nancy Hitschfeld K.

28 octubre 2011

1. Considere la definición de la siguiente jerarquía de clases (c++):

```
class A{
    ...
public:
    A(){
        cout <<"A\n";
    }
    A(int x){
        cout <<"A:"<<x<<"\n";
    }
    virtual ~A(){
        cout <<"~A\n";
    }
    virtual int m()=0;

    virtual void n(){
        cout <<"m="<<m()<<"\n";
    }
    int f(){ return 0;}
};

class B: public A {
    ...
public:
    B(){
        cout <<"B\n";
    }
    B(int x){
        cout <<"B:"<<x<<"\n";
    }
    ~B(){
        cout <<"~B\n";
    }
    virtual int m(){
        return -1;
    }

    int f(){ return 1;}
};

class C: public B{
    ...
public:
    C(){
        cout <<"C\n";
    }
    C(int x){
        cout <<"C:"<<x<<"\n";
    }
    ~C(){
        cout <<"~C\n";
    }
    int m(){
        return 1;
    }
};
```

Para el siguiente código, identifique qué se chequea en tiempo de compilación y qué imprime cada instrucción en tiempo de ejecución. Si hay errores diga por qué se producen. Donde se usa enlace dinámico?

```
A* a = new B();
a->n();
delete a;
B* b = new C();
b->n();
delete b;
```

2. Considere la definición de la siguiente jerarquía de clases (java):

```
class A{
    public A(){
        U.println("A");}
    public A(int x){
        U.println("A"+x);
    }
    public int m(){
        return 0;
    }
    public void n(){
        U.println("n="+m());
    }
}

class B extends A {
    public B(){
        U.println("B");}
    public B(int x){
        U.println("B"+x);
    }
    public int m(){
        return -1;
    }
}

class C extends B{
    public C(){
        U.println("C");}
    public C(int x){
        U.println("C"+x);
    }
    public int m(){
        return 1;
    }
}
```

Para el siguiente código, identifique qué se chequea en tiempo de compilación y qué imprime cada instrucción en tiempo de ejecución. Si hay errores diga por qué se producen. Donde se usa enlace dinámico?

Codigo A

```
A a = new B(2);
a.n();
U.println("m: "+a.m());
B b = new C(5);
b.n();
U.println("m: "+b.m());
```