

Programación Orientada a Objetos

Taller 1

Profesor: Gerardo M. Sarria M.

2 de febrero de 2018

1. Considere el siguiente código:

```
// Prueba1.h
#include <iostream>

using namespace std;

class Papa
{
public:
    void prueba ();
};

class Hijo : public Papa
{
public:
    void prueba ();
};
```

```
// Prueba1.cpp
#include "Prueba1.h"

void Papa::prueba ()
{
    cout << "Clase papa" << endl;
}

void Hijo::prueba ()
{
    cout << "Clase hijo" << endl;
}
```

```
// main.cpp
#include "Prueba1.h"

int main(void)
{
```

```

    Papa* p = new Papa;

    // Imprime "Clase papa"
    p -> prueba ();

    delete p;

    p = new Hijo;

    // Imprime "Clase hijo"
    p -> prueba ();

    delete p;

    return 0;
}

```

Al compilar y ejecutar el código puede darse cuenta que el segundo comentario está equivocado. Modifique el código para que imprima efectivamente “Clase hijo”.

2. Se requieren dos clases, Mamífero y Perro. Perro va a heredar de Mamífero. A continuación se tiene el código de la clase Mamífero. Construya el código para la clase Perro.

```

// Mamifero.h
#include <iostream>

using namespace std;

class Mamifero
{
public:
    Mamifero(void);
    ~Mamifero(void);

    virtual void Mover();
    virtual void Hablar();

protected:
    int edad;
};

```

```

// Mamifero.cpp
#include "Mamifero.h"

Mamifero::Mamifero(void): itsAge(1)
{
    cout << "Mamifero constructor..." << endl;
}

```

```

Mamifero::~~Mamifero(void)
{
    cout << "Mamifero destructor..." << endl;
}

void Mamifero::Mover()
{
    cout << "El mamifero se mueve!" << endl;
}

void Mamifero::Hablar()
{
    cout << "El mamifero habla? Changos!" << endl;
}

```

Una vez haya implementado la clase Perro, compile y ejecute con el siguiente programa principal.

```

#include "Mamifero.h"
#include "Perro.h"

int main ()
{
    Mamifero *perro1 = new Perro;

    perro1 -> Mover();
    perro1 -> Hablar();

    //Perro *perro2 = new Perro;
    //perro2->Mover();
    //perro2->Hablar();

    return 0;
}

```

- ¿Cuál es la salida?
 - Elimine la palabra clave **virtual** de la clase Mamifero y compile y ejecute de nuevo. ¿Cuál es la salida ahora?
 - Descomente el código comentado, compile y ejecute de nuevo. ¿Cuál es la salida final?
 - ¿Qué puede concluir de este ejercicio?
3. Teniendo en cuenta las clases del punto anterior, desarrolle las clases para Gato, Caballo y Cuy sobrescribiendo los métodos Mover y Hablar. Luego pruebe su programa con el siguiente código.

```

int main ()
{

```

```

Mamifero* arreglo [5];
Mamifero* p;
int choice, i;

for (i = 0; i<5; i++)
{
    cout << "(1)perro (2)gato" << endl;
    cout << "(3)caballo (4)cuy: " << endl;
    cin >> choice;

    switch (choice)
    {
        case 1: p = new Perro;
        break;
        case 2: p = new Gato;
        break;
        case 3: p = new Caballo;
        break;
        case 4: p = new Cuy;
        break;
        default: p = new Mamifero;
        break;
    }
    arreglo[i] = p;
}

for (i=0; i<5; i++)
    arreglo[i] -> Hablar();

// Siempre hay que liberar
// la memoria reservada
for (i=0; i<5; i++)
    delete arreglo[i];

return 0;
}

```

4. Después de haber entendido los programas anteriores, responda las siguientes preguntas:

- ¿Los atributos y métodos heredados pasan a las siguientes generaciones? es decir ¿si Perro deriva de Mamífero y Mamífero deriva de Animal, Perro hereda los atributos y métodos de Animal?
- Si, en el ejemplo de arriba, Mamífero sobrescribe un método de Animal, ¿cuál hereda Perro, la original o el método reescrito?
- ¿Puede una clase hijo volver privado un método público heredado?
- ¿Por qué no se hacen todos los métodos de una clase virtuales?
- Si un método (whatever()) es virtual en una clase papa y también es sobrecargado, de manera que acepte un entero o dos enteros como

parámetros, y la clase hijo sobrescribe la forma que toma un entero, ¿es posible hacer el llamado al método con dos enteros por medio de un puntero a un objeto hijo?

5. ¿Cómo se vería el diagrama de clases de un cuadrado, el cual es un rectángulo, el cual es un dibujo, el cual se dibuja con un lápiz?
6. ¿Cuál es el error de las siguientes porciones de código, según el sistema del punto anterior?

```
void unaFuncion ( Dibujo );  
Dibujo * rectangulo = new Rectangulo ;  
unaFuncion (*rectangulo );
```

```
class Dibujo ()  
{  
public :  
    Dibujo ();  
    virtual ~Dibujo ();  
    virtual Dibujo ( Dibujo & );  
};
```