|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2019\_1\_C++ \_10 | 학번 : |  | 이름 : |  |

* **강의 내용**

|  |
| --- |
| 1. 접근지정자에 따른 상속 : private |
| #include <iostream>  using namespace std;  class Base {  int a;  protected:  void setA(int a) { this->a = a; }  public:  void showA() { cout << a; }  };  class Derived : private Base {  int b;  protected:  void setB(int b) { this->b = b; }  public:  void showB() { cout << b; }  };  int main() {  Derived x;  x.a = 5; // ①  x.setA(10); // ②  x.showA(); // ③  x.b = 10; // ④  x.setB(10); // ⑤  x.showB(); // ⑥  } |
| **[실행결과]** |

|  |
| --- |
| 1. 접근지정자에 따른 상속 : protected |
| #include <iostream>  using namespace std;  class Base {  int a;  protected:  void setA(int a) { this->a = a; }  public:  void showA() { cout << a; }  };  class Derived : protected Base {  int b;  protected:  void setB(int b) { this->b = b; }  public:  void showB() { cout << b; }  };  int main() {  Derived x;  x.a = 5; // ①  x.setA(10); // ②  x.showA(); // ③  x.b = 10; // ④  x.setB(10); // ⑤  x.showB(); // ⑥  } |
| **[실행결과]** |

|  |
| --- |
| 1. 함수 재정의 : |
| **class Base** {  public:  **void f()** { cout << "Base::f() called" << endl; }  };  **class Derived : public Base** {  public: **//함수 재정의**  **void f()** { cout << "Derived::f() called" << endl; }  };  void main() {  **Derived d**, \*pDer;  pDer = &d;  **pDer->f(); // Derived::f() 호출**  **//pDer->Base::f() …. Base::f() 호출**    Base\* pBase;  pBase = pDer; // 업캐스팅  **pBase->f(); // Base::f() 호출**  } |
| **[실행결과]** |

|  |
| --- |
| 1. 가상 함수와 오버라이딩 |
| **======= Circle.h =====**  #ifndef CIRCLE  #define CIRCLE  class Circle : public Shape {  protected:  virtual void draw();  };  #endif  **======= Line.h ====**  #ifndef LINE  #define LINE  class Line : public Shape {  protected:  virtual void draw();  };  #endif  **======== Rect.h =====**  #ifndef RECT  #define RECT  class Rect : public Shape {  protected:  virtual void draw();  };  #endif  **=======Shape.h ======**  #ifndef SHAPE  #define SHAPE  class Shape {  Shape\* next;  protected:  virtual void draw();  public:  Shape() { next = NULL;}  virtual ~Shape() { }  void paint();  Shape\* add(Shape\* p);  Shape\* getNext() { return next;}  };  #endif  **======== Circle.cpp =====**  #include <iostream>  #include "Shape.h"  #include "Circle.h"  using namespace std;  void Circle::draw() {  cout << "Circle" << endl;  }  **===== Line.cpp ====**  #include <iostream>  #include "Shape.h"  #include "Line.h"  using namespace std;  void Line::draw() {  cout << "Line" << endl;  }  =====Rect.cpp ====  #include <iostream>  #include "Shape.h"  #include "Rect.h"  using namespace std;  void Rect::draw() {  cout << "Rectangle" << endl;  }  **=== Shape.cpp ===**  #include <iostream>  #include "Shape.h"  using namespace std;  void Shape::paint() {  draw();  }  void Shape::draw() {  cout << "Shape" << endl;  }  Shape\* Shape::add(Shape \*p) {  this->next = p;  return p;  }  **==== main.cpp ====**  #include <iostream>  #include "Shape.h"  #include "Circle.h"  #include "Rect.h"  #include "Line.h"  using namespace std;  int main() {  Shape \*pStart=NULL;  Shape \*pLast;  pStart = new Circle(); // 처음에 원 도형을 생성한다.  pLast = pStart;  pLast = pLast->add(new Rect()); // 사각형 객체 생성  pLast = pLast->add(new Circle()); // 원 객체 생성  pLast = pLast->add(new Line()); // 선 객체 생성  pLast = pLast->add(new Rect()); // 사각형 객체 생성  // 현재 연결된 모든 도형을 화면에 그린다.  Shape\* p = pStart;  while(p != NULL) {  p->paint();  p = p->getNext();  }  // 현재 연결된 모든 도형을 삭제한다.  p = pStart;  while(p != NULL) {  Shape\* q = p->getNext(); // 다음 도형 주소 기억  delete p; // 기본 클래스의 가상 소멸자 호출  p = q; // 다음 도형 주소를 p에 저장  }  } |
| **[실행결과]** |

|  |
| --- |
| 1. 가상 함수와 오버라이딩 |
| #include <iostream>  using namespace std;  class Base {  public:  **virtual ~Base()** { cout << "~Base()" << endl; }  };  class Derived: public Base {  public:  **virtual ~Derived()** { cout << "~Derived()" << endl; }  };  int main() {  Derived \*dp = new Derived();  Base \*bp = new Derived();  **delete dp;** // Derived의 포인터로 소멸  **delete bp;** // Base의 포인터로 소멸  } |
| **[실행결과]** |

* 내용 점검

1. 교재 8장 연습문제

|  |  |
| --- | --- |
| 번호 | 문제풀이 |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 | (1)  (2) |
| 8 | (1)  (2)  (3)  (4) |
| 9 | (1)  (2)  (3) |
| 10 | (1)  (2)  (3) |
| 11 |  |
| 12 |  |
| 13 |  |
| 14 | (1)  (2) |
| 15 |  |
| 16 |  |

* 프로그램 과제

|  |
| --- |
| 1. 교재 420p : 문제5번 |
| **[프로그램 소스]** |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 421p : 문제6번 |
| **[프로그램 소스]** |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 422p : 문제8번 |
| **[프로그램 소스]** |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 423p 문제9번 |
| **[프로그램 소스]** |
| **[실행결과]** |