**객체지향프로그래밍 LAB #03&04**

**<기초문제>\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. 아래의 프로그램을 작성하시오. (/\*구현\*/ 부분을 채울 것, 표의 상단: 소스코드, 하단: 실행결과)

square 함수 : 입력의 제곱을 반환하는 함수

|  |
| --- |
| /\*구현\*/ //함수 선언  int main() {  int x;  x = 5;  cout << /\*구현\*/ << endl; // square 함수 사용  }  //함수 정의 |
|  |
| #include <iostream>  using namespace std;  int square(int param);  int main() {  int x;  x = 5;  cout << square(x) << endl;    return 0;  }  int square(int param) {  return param \* param;  } |

텍스트이(가) 표시된 사진

자동 생성된 설명

2. 아래의 프로그램을 작성하시오. (/\*구현\*/ 부분을 채울 것)

|  |
| --- |
| int get\_num();  int myadd(int x, int y);  int mysub(int x, int y);  int mymul(int x, int y);  float mydiv(int x, int y);  int main() {  int x, y;  x = /\*구현\*/ //키보드로부터 값 입력  y = /\*구현\*/  cout << "x = " << x << ", y = " << y << endl;  cout << "x + y = " << /\*구현\*/ << endl;  cout << "x - y = " << /\*구현\*/<< endl;  cout << "x \* y = " << /\*구현\*/ << endl;  cout << "x / y = " << /\*구현\*/ << endl;  return 0;  }  int get\_num() {  int num;  cout << "Enter a number: ";  cin >> num;  return num;  }  int myadd(int x, int y) {  return /\*구현\*/ }  int mysub(int x, int y) {  return /\*구현\*/ }  int mymul(int x, int y) {  return /\*구현\*/ }  float mydiv(int x, int y) {  return /\*구현\*/ //소수점 출력되도록 } |
|  |
| #include <iostream>  using namespace std;  int get\_num();  int myadd(int x, int y);  int mysub(int x, int y);  int mymul(int x, int y);  float mydiv(int x, int y);  int main() {  int x, y;  x = get\_num();  y = get\_num();  cout << "x = " << x << ", y = " << y << endl;  cout << "x + y = " << myadd(x, y) << endl;  cout << "x - y = " << mysub(x, y) << endl;  cout << "x \* y = " << mymul(x, y) << endl;  cout << "x / y = " << mydiv(x, y) << endl;  return 0;  }  int get\_num() {  int num;  cout << "Enter a number: ";  cin >> num;  return num;  }  int myadd(int x, int y) {  return x + y;  }  int mysub(int x, int y) {  return x - y;  }  int mymul(int x, int y) {  return x \* y;  }  float mydiv(int x, int y) {  return (float)x / (float)y;  } |

텍스트이(가) 표시된 사진

자동 생성된 설명

3. 아래의 프로그램을 작성하시오. (/\*구현\*/ 부분을 채울 것)

|  |
| --- |
| int main() {  double value;  value = 5;  cout << /\*구현\*/<< endl; //제곱근  cout << /\*구현\*/<< endl; //e의 지수 값 연산  cout << /\*구현\*/ << endl; //로그연산(밑:10)  cout << /\*구현\*/ << endl; //cosine  return 0;  } |
|  |
| #include <iostream>  #include <cmath>  using namespace std;  int main() {  double value;  value = 5;  cout << sqrt(value) << endl;  cout << exp(value) << endl;  cout << log(value) << endl;  cout << cos(value) << endl;  return 0;  } |

텍스트이(가) 표시된 사진

자동 생성된 설명

4. 아래의 프로그램을 작성하시오. (/\*구현\*/ 부분을 채울 것)

|  |
| --- |
| void applePrice(int a = 1000) {  cout << "Price of an apple is " << a << endl;  }  // default augment 중요!  int main() {  /\*구현\*/  /\*구현\*/  /\*구현\*/  /\*구현\*/ // default augments  return 0;  } |
|  |
| #include <iostream>  using namespace std;  void applePrice(int a = 1000) {  cout << "Price of an apple is " << a << endl;  }  int main() {  applePrice(1500);  applePrice(500);  applePrice(1000);  applePrice(); // default augments  return 0;  } |

**텍스트이(가) 표시된 사진

자동 생성된 설명**

**<응용문제>\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. 기초문제 2번 문제에서 구현한 4개의 함수 (myadd, mysub, mymul, mydiv) 만을 사용하여, 복합적인 연산을 수행하는 아래의 함수들을 구현하시오.

* addmul(int x, int y, int z) // return value: (x + y) \* z
* muldiv(int x, int y, int z) // return value: (x \* y) / z
* addmuladd(int x, int y, int z) // return value: (x + y) \* (y + z)
* subdivsub(int x, int y, int z) // return value: (x - y) / (y - z)

1-출력화면:

텍스트이(가) 표시된 사진

자동 생성된 설명

|  |
| --- |
| #include <iostream>  using namespace std;  int get\_num() {  int num;  cout << "Enter a number: ";  cin >> num;  return num;  }  int myadd(int x, int y) {  return x + y;  }  int mysub(int x, int y) {  return x - y;  }  int mymul(int x, int y) {  return x \* y;  }  float mydiv(int x, int y) {  return (float)x / (float)y;  }  int addmul(int x, int y, int z) {  return mymul(myadd(x, y), z);  }  float muldiv(int x, int y, int z) {  return mydiv(mymul(x, y), z);  }  int addmuladd(int x, int y, int z) {  return mymul(myadd(x, y), myadd(y, z));  }  float subdivsub(int x, int y, int z) {  return mydiv(mysub(x, y), mysub(y, z));  }  int main() {  int x, y, z;  x = get\_num();  y = get\_num();  z = get\_num();  cout << "x = " << x << ", y = " << y << ", z = " << z << endl;  cout << "(x + y) \* z = " << addmul(x, y, z) << endl;  cout << "(x \* y) / z = " << muldiv(x, y, z) << endl;  cout << "(x + y) \* (y + z) = " << addmuladd(x, y, z) << endl;  cout << "(x - y) / (y - z) = " << subdivsub(x, y, z) << endl;  return 0;  } |

텍스트이(가) 표시된 사진

자동 생성된 설명

2. 다음 조건을 만족하는 함수를 구현하시오.

* print\_DOB 함수는 정수형 year, month, day 3개의 파라미터를 입력으로 받는다.
* print\_DOB 함수는 default augments를 설정하여 print\_DOB 함수의 입력 값이 없을 경우 아래 화면과 같이 기본 출력 값이 나온다.
* 함수의 입력 값이 있으면 아래 화면과 같이 생년월일을 출력한다.

|  |
| --- |
| **입출력 예시:**   * 입력 값이 없을 경우   (output) 생년월일은 1900년 1월 1일입니다.   * 입력 값이 있을 경우   (input) 1998  9  8  (output) 생년월일은 1998년 9월 8일입니다. |

※ 입력 값이 없을 경우 출력되는 default augments를 확인하기 위해 print\_DOB 함수 먼저 실행함.

2-출력화면:

텍스트이(가) 표시된 사진

자동 생성된 설명

|  |
| --- |
| #include <iostream>  using namespace std;  void print\_DOB(int year = 1900, int month = 1, int day = 1) {  cout << "생년월일은 " << year << "년 " << month << "월 " << day << "일입니다." << endl;  }  int main() {  int year, month, day;  print\_DOB();  cout << "year 입력 : ";  cin >> year;  cout << "month 입력 :";  cin >> month;  cout << "day 입력 : ";  cin >> day;  print\_DOB(year, month, day);  return 0;  } |

텍스트이(가) 표시된 사진

자동 생성된 설명

3. 두 개의 double 형 실수(x, y)를 키보드로 입력 받아 cmath 라이브러리를 이용하여 아래의 값을 계산 후 화면에 출력하시오. (참고: <http://www.cplusplus.com/reference/cmath/>)

* 를 올림한 값
* 를 버림한 값
* 를 반올림한 값
* 와 중에서 큰 값
* 제곱근 (과 같음.)
* 와 의 차이 (절대값)

3-출력화면:



|  |
| --- |
| #include <iostream>  #include <cmath>  using namespace std;  int main() {  double x, y;  cout << "x = ";  cin >> x;  cout << "y = ";  cin >> y;  double minus = x - y;  double c = 1 / y;  cout << "Rounding up number of " << x << ": " << ceil(x) << endl;  cout << "Rounding down number of " << x << ": " << floor(x) << endl;  cout << "Rounding number of " << x << ": " << round(x) << endl;  cout << "Maximum number between " << x << " and " << y << ": " << max(x,y) << endl;  cout << y << "-th root of " << x << ": " << pow(x, c) << endl;  cout << "|" << x << " - " << y << "| = " << fabs(minus) << endl;  return 0;  } |

텍스트이(가) 표시된 사진

자동 생성된 설명

4. Call by value와 Call by reference를 사용하여 swap함수를 구현하시오. Call by value와 Call by reference의 차이를 보기 위해 아래의 출력화면과 같이 두 swap 함수 사용 전, 후의 x, y값을 출력하시오.

* Call by value를 사용한 swap 함수의 이름을 swap\_call\_by\_value로 할 것.
* Call by reference를 사용한 swap 함수의 이름을 swap\_call\_by\_reference로 할 것.
* 키보드로부터 정수 x, y를 입력 받는 get\_data(int& x, int& y)를 구현할 것. main함수에서는 get\_data함수를 통해 두 정수 x, y를 키보드로부터 입력 받음.

4-출력화면:



|  |
| --- |
| #include <iostream>  using namespace std;  int get\_data(int& x, int& y) {  cout << "x 입력 : ";  cin >> x;  cout << "y 입력 : ";  cin >> y;  return x, y;  }  void swap\_call\_by\_value(int a, int b) {  int temp = a;  a = b;  b = temp;  }  void swap\_call\_by\_reference(int& a, int& b) {  int temp = a;  a = b;  b = temp;  }  int main() {  int x, y;  get\_data(x, y);  cout << endl;  cout << "swap\_call\_by\_value 함수 사용 전" << endl;  cout << "x = " << x << ", y = " << y << endl;  swap\_call\_by\_value(x, y);  cout << "swap\_call\_by\_value 함수 사용 후" << endl;  cout << "x = " << x << ", y = " << y << endl;  cout << endl;  cout << "swap\_call\_by\_reference 함수 사용 전" << endl;  cout << "x = " << x << ", y = " << y << endl;  swap\_call\_by\_reference(x, y);  cout << "swap\_call\_by\_reference 함수 사용 후" << endl;  cout << "x = " << x << ", y = " << y << endl;  return 0;  } |

텍스트이(가) 표시된 사진

자동 생성된 설명