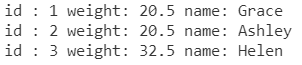
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| cpp\_11주 | 학번 : |  | 이름 : |  |

* **개념 적용 응용 프로그래밍 – 함수 중복과 static 멤버**

1. 제시된 main()함수와 결과를 보고 디폴트 매개변수를 가진 하나의 생성자를 갖는 Person 클래스를 작성하고 프로그램을 완성 하시오



int main() {

  Person grace, ashley(2, "Ashley"), helen(3, "Helen", 32.5); //id, name

  grace.show();

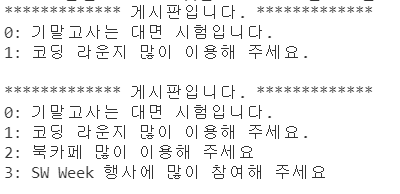
  ashley.show();

  helen.show();

}

|  |
| --- |
| [프로그램 소스] |
| [실행 결과] |

1. 실행 결과와 main()을 참고하여 Board 클래스를 작성하고 프로그램을 완성하시오



int main() {

  Board::add("기말고사는 대면 시험입니다.");

  Board::add("코딩 라운지 많이 이용해 주세요.");

  Board::print();

  Board::add("북카페 많이 이용해 주세요");

  Board::add("SW Week 행사에 많이 참여해 주세요");

  Board::print();

}

|  |
| --- |
| [프로그램 소스] |
| [실행 결과] |

1. 큰 값을 리턴하는 big 함수를 중복 구현하라

class Big {

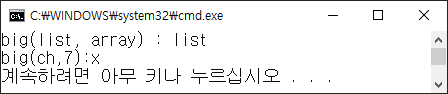
public:

string big(string a, string b); // a와 b 중 큰 문자열 리턴

int big(char a[], int size); // 배열 a에 저장된 문자 중 아스키 코드 값이 큰 문자의 인덱스 리턴

};

char ch[] = { 'd', 'a', 'e', 'A', 'p', 'x', 'q' };



|  |
| --- |
| [프로그램 소스] |
| [실행 결과] |

1. 디폴트 매개변수를 갖는 하나의 add() 함수를 작성하고 프로그램을 완성 하시오

int main() {

int a[] = {1,2,3,4,5};

int b[] = {6,7,8,9,10};

int c = add(a, 5); // 배열 a의 정수를 모두 더한 값 리턴

int d = add(a, 5, b); // 배열 a와 b의 정수를 모두 더한 값 리턴

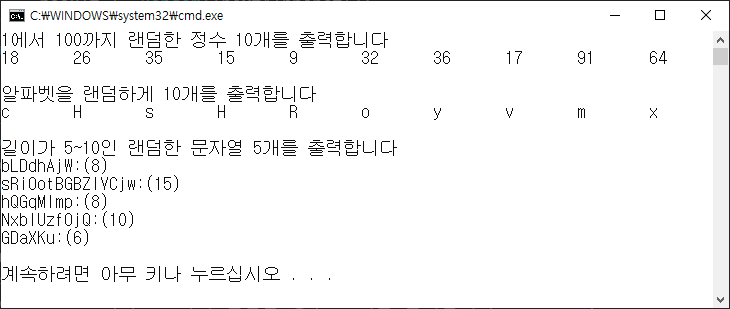
cout << c << endl; // 10 출력

cout << d << endl; // 55 출력

}

|  |
| --- |
| [프로그램 소스] |
| [실행 결과] |

1. 다음과 같은 static 멤버를 갖는 Random 클래스를 완성하고 제시된 결과처럼 출력될 수 있도록 main()함수도 작성하시오



class Random {

public:

static int nextInt(int min = 0, int max = 32767); //min과 max 사이의 랜덤 정수 리턴

static char nextAlphabet(); //랜덤하게 영문자 반환

static string nextString(int length); //매개변수로 전달된 길이를 갖는 임의의 문자열 반환

};

|  |
| --- |
| [프로그램 소스] |
| [실행 결과] |

1. 다음과 같이 동일한 크기를 갖는 두 개의 배열을 매개변수로 전달하여 제시된 결과처럼 실행하는 프로그램을 완성 하시오.

class ArrayUtility2 {

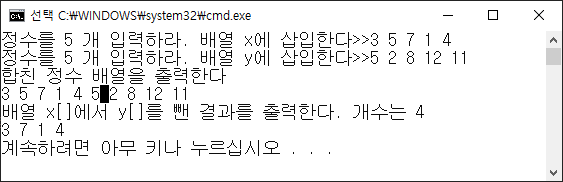
public:

static int\* concat(int s1[], int s2[], int size); // s1과 s2를 연결한 새로운 배열을 동적 생성하고 포인터 리턴

// s1에서 s2에 있는 숫자를 모두 삭제한 새로운 배열을 동적 생성하여 리턴

static int\* remove(int s1[], int s2[], int size, int& retSize);

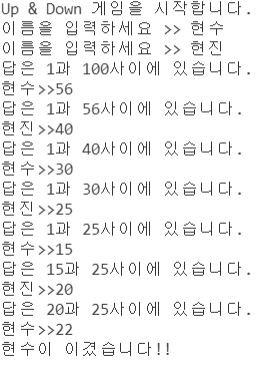
};



|  |
| --- |
| [프로그램 소스] |
| [실행 결과] |

1. 2명의 사람이 번갈아 숨겨진 답을 찾는 사람이 이긴다. 찾아야 할 답은 난수로 생성하며 static 멤버로 구성하는 UpAndDownGame 클래스와 선수를 다루기 위한 Person 클래스를 구현하여 처리하도록 한다.

class Person {

 string name;

public:

Person(string name) { this->name = name; }

string getName() { return name; }

bool go(); // 정수를 입력 받고 정답이면 승리, true 리턴

};

class UpAndDownGame {

static int answer; //맞춰야 할 답, 난수로 초기화

static int top; //맞춰야 할 답의 범위에서 최고값

static int bottom; //맞춰야 할 답의 범위에서 최저값

public:

static void run(); // 게임 진행

static bool check(int answer); //top과 bottom을 조절하고, 정답을 맞추었으면 true 리턴

};

int main() {

UpAndDownGame::run();

}

|  |
| --- |
| [프로그램 소스] |
| [실행 결과] |