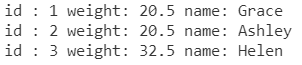
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| cpp\_11주 | 학번 : | 20175260 | 이름 : | 최영빈 |

* **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();

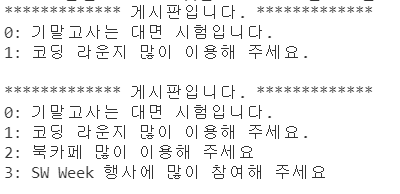
  ashley.show();

  helen.show();

}

|  |
| --- |
| [프로그램 소스]   #include <iostream>  #include <string>  using namespace std;  class Person{  int id;  double weight;  string name;  public :  Person(int i = 1, string n = "Grace", double w = 20.5);  void show();  };  Person::Person(int i, string n, double w)  {  id = i;  weight = w;  name = n;  }  void Person::show(){  cout << "id : " <<id << " weight : "<< weight <<" name : " << name <<endl;  }  int main()  {  Person grace, ashley(2, "Ashley"), helen(3, "Helen", 32.5); // id, name  grace.show();  ashley.show();  helen.show();  } |
| [실행 결과]  이미지 |

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]   #include <iostream>  #include <string>  using namespace std;  class Board{  inline static int size = 0;  inline static string text[100];  public:  Board() = delete;  static void add(string &&text);  static void print();  };  void Board::add(string &&t){  if(size > 99)  return;  text[size] =t;  size++;  }  void Board::print(){  cout << "\*\*\*\*\* 게시판 입니다. \*\*\*\*\*"<<endl;  for(int i =0; i < size; i++){  cout <<i <<": "<<text[i] <<endl;  }  cout <<endl;  }  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();  } |
| [실행 결과]  이미지 |

1. 큰 값을 리턴하는 big 함수를 중복 구현하라

class Big {

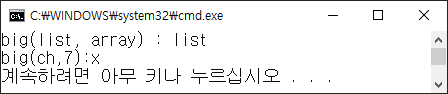
public:

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]  #include <iostream>  #include <string>  using namespace std;  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  };  string Big::big(string a, string b){  if(a>b)  return a;  return b;  }  int Big::big(char a[], int size){  int max = 0;  for(int i =1; i <size; i++){  if(a[max] < a[i]){  max = i;  }  }  return max;  }  char ch[] = {'d', 'a', 'e', 'A', 'p', 'x', 'q'};  int main()  {  Big a;  int result = a.big(ch,7);  cout << a.big("list ","array") <<endl;  cout << ch[result] <<endl;    } |
| [실행 결과]  이미지 |

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

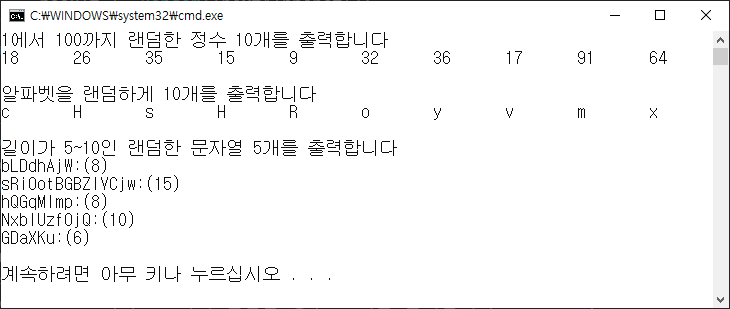
cout << c << endl; // 10 출력

cout << d << endl; // 55 출력

}

|  |
| --- |
| [프로그램 소스]   #include <iostream>  #include <string>  using namespace std;  int add(int \*a, int b, int \*c = nullptr);  int add(int \*a, int b, int \*c){  int sum = 0;  for(int i = 0; i <b; i++){  sum += a[i];  }  if(c != nullptr){  for(int i =0; i < b; i++){  sum += c[i];  }  }  return sum;  }  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  } |
| [실행 결과]  이미지 |

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]   #include <iostream>  #include <string>  #include <ctime>  #include <cstdlib>  #include <iomanip>  using namespace std;  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  };  int Random::nextInt(int min, int max){  return rand()%max;  }  char Random::nextAlphabet(){  return static\_cast<char>(65 + rand() % 26);  }  string Random::nextString(int length){  string result = "";  int count = 0;  for(int i =0; i < length; i++){  result += static\_cast<char>(65+rand()%26);  }  return result;  }  int main()  {  srand(time(nullptr));  Random r;  cout << "1에서 100까지 랜덤한 정수 10개를 출력합니다"<<endl;  for(int i =0 ; i < 10; i ++){  cout<<r.nextInt(0,100)<<setw(4);  }  cout <<endl<< "알파벳을 랜덤하게 10개를 출력합니다."<<endl;  for(int i =0; i <10; i++){  cout <<r.nextAlphabet()<<setw(4);  }  cout <<endl<<"길이가 5~10인 랜덤한 문자열 5개를 출력합니다"<<endl;  for(int i =0; i < 5; i++){  int size = 5+rand()%5;  cout<<r.nextString(size)<<": (" << size <<" )"<<endl;  }  } |
| [실행 결과]  이미지 |

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

class ArrayUtility2 {

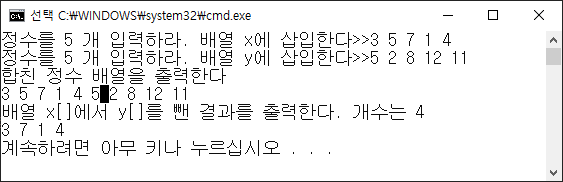
public:

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);

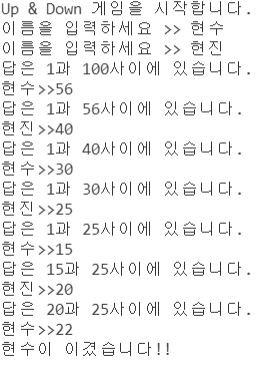
};



|  |
| --- |
| [프로그램 소스]  #include <iostream>  #include <string>  using namespace std;  class ArrayUtility2  {  int \*arr1;  int \*arr2;  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);  };  int\* ArrayUtility2::concat(int s1[],int s2[],int size){  int \*arr1 = new int[size];  for(int i =0; i <5; i++){  arr1[i] = s1[i];  }  for(int i =5; i <10; i++){  arr1[i] = s2[i%5];  }  return arr1;  }  int\* ArrayUtility2::remove(int s1[],int s2[],int size, int &retSize){  int \*arr2 = new int[retSize];  for(int i =0; i <retSize; i++){  int count = s1[i] - s2[i];  if(count < 0)  count \*= -1;  arr2[i] = count;  }  return arr2;  }  int main()  {  ArrayUtility2 a;  int b[5];  int c[5];  int size = 0;  cout <<"정수를 5개 입력하라. 배열 x에 삽입한다 >>";  for(int i = 0; i < 5; i++){  cin >> b[i];  }  cout << "정수를 5개 입력하라. 배열 y에 삽입한다 >>";  for (int i = 0; i < 5; i++)  {  cin >> c[i];  }  cout <<"합친 정수 배열을 출력한다 "<<endl;  int \*result = a.concat(b,c,10);  for(int i = 0; i < 10; i++){  cout <<result[i]<<" ";  }  cout <<endl <<"배열 x[] 에서 y[]를 뺀 결과를 출력한다." << "개수는 ";  cin >>size;  result = a.remove(b,c,5,size);  for(int i =0; i< size; i++){  cout <<result[i]<<" ";  }  cout <<endl;  delete[] result;  a.~ArrayUtility2();  } |
| [실행 결과]  이미지 |

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]   #include <iostream>  #include <string>  #include <ctime>  #include <cstdlib>  using namespace std;  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 UpAndDownGame::top =100;  int UpAndDownGame::bottom = 1;  int UpAndDownGame::answer = 0;  bool Person::go()  {  int result;  cout << name << " >> ";  cin >> result;  bool f = UpAndDownGame::check(result);  return f;  }  void UpAndDownGame::run(){  srand(time(nullptr));  string a;  string b;  string result;  answer =rand() % 100;  bool flag = false;  cout << "Up & Down 게임을 시작합니다. " << endl;  cout << "이름을 입력하세요 >> ";  cin >> a;  cout << "이름을 입력하세요 >> ";  cin >> b;  Person p1(a);  Person p2(b);  while(!flag){  cout << "답은 " << bottom << "과 " << top << "사이에 있습니다." << endl;  result = a;  flag = p1.go();  if(flag)  break;  cout << "답은 " << bottom << "과 " << top << "사이에 있습니다." << endl;  result = b;  flag = p2.go();  }  if(result == a){  cout<<p1.getName()<<"이 이겼습니다."<<endl;  }  else  cout << p2.getName() << "이 이겼습니다." << endl;  }  bool UpAndDownGame::check(int a){  if(answer == a){  return true;  }  else{  if(answer > a)  bottom = a;  else  top = a;  }  return false;  }  int main()  {  UpAndDownGame::run();  } |
| [실행 결과]  이미지 |