

1. FirstStep.cpp

FirstStep.cpp

```
1  #include    <iostream>
2
3  int main()
4  {
5      std::cout << "나의 첫 번째 C++ 프로그램" << std::endl;
6      return 0;
7  }
```

5행을 다음과 같이 작성할 수 있도록 수정하라.

```
cout << "나의 첫 번째 C++ 프로그램" << endl;
```

2. 10개의 값이 저장된 배열에서 최댓값 구하기

다음 프로그램의 (㉠)~(㉤)에 넣을 내용을 작성하라.

MaxValue.cpp

```
#include <iostream>
using namespace std;

int main()
{
    int data[10] = { 10, 23, 5, 9, 22, 48, 12, 10, 55, 31 };
    int (㉠); // data의 첫 번째 데이터를 max로 가정함

    cout << "데이터 : " << (㉡); // 0번 데이터 출력
    for ( (㉢) ) { // 나머지 9개의 데이터 비교
        cout << " " << (㉣); // i번 데이터 출력
        if (max < (㉤)) // i번 데이터가 max보다 크면 비교
            max = (㉤); // max를 i번 데이터로 바꿈
    }
    cout << endl << endl;
    cout << "배열의 최댓값 : " << max << endl;
    return 0;
}
```

3. 포인터 예제

(가) 다음 프로그램의 (㉠)~(㉢)에 넣을 내용을 작성하라.

(나) 이 프로그램의 출력 결과를 구하라.

IntPtr.cpp

```
#include <iostream>
using namespace std;

int main()
{
    int a = 10, b = 100;
    int (㉠); // 포인터 ptr을 선언한 후에 a의 주소를 넣는다.

    cout << "ptr이 가리키는 곳의 값 : " << (㉡) << endl;
    (㉢) = 20; // ptr이 가리키는 곳에 20을 저장
    cout << "변수 a의 값 : " << a << endl;
    (㉣); // ptr이 b를 가리키게 함
    cout << "변수 b의 값 : " << *ptr << endl;
    return 0;
}
```

4. 동적 메모리 할당 예제

(가) 다음 프로그램의 (㉠), (㉡), (㉢)에 넣을 문장을 작성하라.

(나) 이 프로그램의 출력 결과를 구하라.

DAlloc.cpp

```
#include <iostream>
using namespace std;

int main()
{
    int *intPtr;
    (㉠) // 4개의 int 값을 저장할 메모리 할당
    *intPtr = 10;
    *(intPtr + 1) = 20;
    intPtr[2] = 30;
    intPtr[3] = 40;
    for (int *p = intPtr, i = 0; i < 4; i++)
        cout << (㉡) << " "; // p가 가리키는 곳의 값을 출력한 후 p를 다음으로 이동
    cout << endl;
    (㉢) // (㉠)에서 할당한 메모리 반환
    return 0;
}
```

5. 참조 활용 예제

(가) (㉠)에 넣을 문장을 작성하라

(나) 이 프로그램의 출력 결과를 구하라.

RecTest.cpp

```
#include <iostream>
using namespace std;

int main()
{
    int a = 10, b = 20;
    (㉠) // a를 참조하는 참조 변수 aRef를 선언
    cout << "a의 값 : " << a << endl;
    cout << "aRef가 참조하는 값 : " << aRef << endl << endl;
    aRef = 100;
    cout << "a의 값 : " << a << endl;
    aRef = b;
    cout << "a의 값 : " << a << endl;
    return 0;
}
```

6. 함수 활용 예제

(가) (ㄱ)에 넣을 형식 매개변수 목록을 작성하라.

- int형 배열 arr과 배열의 크기를 나타내는 정수형 매개변수 len

(나) (ㄴ)에 넣을 문장을 작성하라.

(다) 이 프로그램의 출력 결과를 구하라.

(라) float형 배열에 사용할 수 있도록 getMax 함수를 다중정의하라.

GetMax.cpp

```
#include <iostream>
using namespace std;

int getMax( (ㄱ) ) // 함수 머리부
{
    int max = INT_MIN; // int형의 최솟값을 max로 가정함
    for (int i = 0; i < len; i++)
        if (max < arr[i]) max = arr[i];
    (ㄴ) // 결과의 반환
}

int main()
{
    int data[10] = { 10, 23, 5, 9, 22, 48, 12, 10, 55, 31 };

    cout << "데이터 : ";
    for (int i = 0; i < 10; i++)
        cout << data[i] << " ";
    cout << endl << endl;
    cout << "최댓값 = " << getMax(data, 10) << endl;
}
```

7. 참조호출

(㉠)~(㉢)에 적절한 내용을 작성하라.

CallByRef.cpp

```
#include <iostream>
using namespace std;

const float PI = 3.14159265;
struct Circle {
    float radius, cx, cy;
};

// 원의 데이터 입력
(㉠) inputData( (㉡) ) // 함수 머리부
{
    cout << "원의 반지름 : ";
    cin >> c.radius;
    cout << "중심좌표(x) : ";
    cin >> c.cx;
    cout << "중심좌표(y) : ";
    cin >> c.cy;
}

// 원의 데이터 출력
(㉢) prData( (㉣) ) // 함수 머리부
{
    cout << "반지름 = " << c.radius << endl;
    cout << "중심좌표 = (" << c.cx;
    cout << ", " << c.cy << ")" << endl;
}

(㉤) area( (㉥) )
{
    return PI * c.radius * c.radius;
}

int main()
{
    Circle circle = {1, 2, 3};
    inputData(circle);
    cout << "입력된 원의 정보" << endl;
    prData(circle);
    cout << "원의 면적 = " << area(circle) << endl;
}
```

8. 함수 다중정의 예제

예제7의 함수 inputData, prData, area를 직사각형을 표현하는 다음 구조체에 사용할 수 있도록 다중정의하라.

```
struct Rectangle {  
    float x1, y1; // 좌측 하단의 꼭짓점 좌표  
    float x2, y2; // 우측 상단의 꼭짓점 좌표  
};
```

