2장 제어의 흐름

2.1 부울 식

- 부울(bool, 논리) 식은 참 또는 거짓 값을 가지는 모든 식이다.

예: 5<6, x > y

* 부울 식의 구성

- 단일 부울 식을 논리 및 비교 연산자를 사용하여 확장할 수 있다.

|  |  |  |  |
| --- | --- | --- | --- |
| 연산자 | 의미 | C++ 예제 | 수학적 표현 |
| && | and | (2<x) && (x<7) | (2<x) and (x<7) |
| || | or | (y <0) || (y>12) | (y <0) or (y>12) |
| == | = | (x+7) == (2\*y) | (x+7) = (2y) |
| <= | ≤ | time <= limit | time ≤ limit |
| !  (사용자제) | not | !(x<y) |  |
| 좋은 방법 | (x>=y) | (x≥y) |

- bool 변수는 부울 식의 결과 값을 true 또는 false로 저장할 수 있다.

참: 1byte 값이 정확히 1일 때, 거짓: 1이외의 값은 모두 거짓이다.

cout.setf(ios::boolalpha ); bool 변수를 “true, false”로 출력

cout.unsetf(ios::boolalpha); bool 변수를 다시 1과 0으로 출력

* 우선순위

- not 연산자(!), 관계 연산자(<, >, ==), 논리 연산자(&&, ||) 순서로 실행된다.

- 순위가 동일한 경우, 단항 연산자는 오른쪽에서 왼쪽으로, 이항 연산자는 왼쪽에서 오른쪽으로 실행된다.

- 혼동을 피하기 위해서는 **괄호를 사용하라**.

정수 값도 부울 값으로 사용할 수 있다.

- 0을 false로, 0이 아닌 모든 **정수**를 true로 인식한다. 컴파일러에 따라서는 정수 1만 true로 인식하는 경우도 있다.

단락회로 평가(short-circuit evaluation)

- &&나 ||에서 첫 번째 식을 먼저 계산하여 그 값이 전체 식의 값을 결정하기에 충분하다면 두 번째 식을 계산하지 않고 전체 식을 결정한다.

예: (5>6) && (x > y)

예제: 부울 식

ex02\_01.cpp

|  |
| --- |
| #include <iostream>  using namespace std;  int main()  {  int score;  bool decision;  cout.setf(ios::boolalpha);  cout << "Enter your score: ";  cin >> score;  cout << (score >= 60) << endl;  decision = (score >= 60);  cout << decision << endl;  cout << !(false || true) << endl;  cout << ((5 > 6) && (7 / 0)) << endl;  cout << !(5 && 7) << !(!6) << endl;  return 0;  } |

2.2 분기 메커니즘

* if-else 문

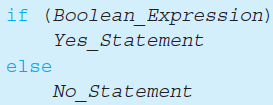
- 부울 식은 반드시 괄호로 감싸야 한다.

- else 문은 선택 사항이다.

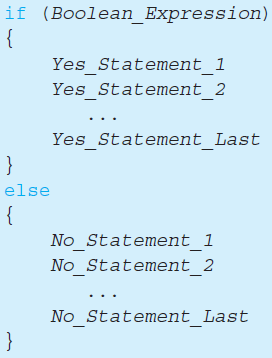
1개 실행문: if 문



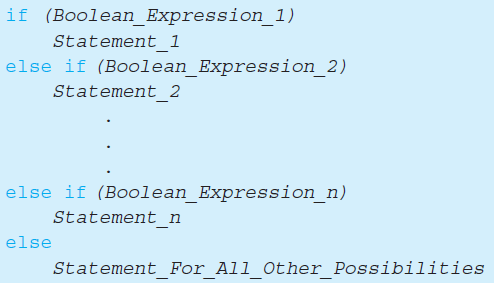
2개 실행문: if-else 문



복합 실행문(실행문 덩어리): 중괄호{}로 묶는다.



다중 if-else 문: 참 값이 나올 때까지 수행한다.



예제: if-else 문

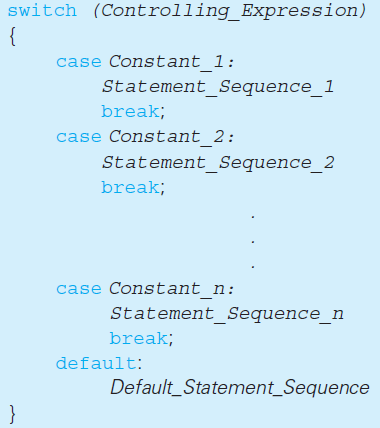
ex02\_02.cpp

|  |
| --- |
| #include <iostream>  using namespace std;  int main()  {  int score;  char grade;  cout << "Enter your score: ";  cin >> score;  //if (score >= 60) cout << "You passed" << endl;  //if (score < 60) cout << "You failed" << endl;  //if (score >= 60) cout << "You passed" << endl;  //else cout << "You failed" << endl;  /\*if (score >= 60)  {  grade = 'P';  cout << "Your grade is " << grade << endl;  }  else  {  grade = 'F';  cout << "Your grade is " << grade << endl;  }\*/  if (score >= 90)  {  grade = 'A';  cout << "Your grade is " << grade << endl;  }  else if (score >= 80)  {  grade = 'B';  cout << "Your grade is " << grade << endl;  }  else if (score >= 70)  {  grade = 'C';  cout << "Your grade is " << grade << endl;  }  else if (score >= 60)  {  grade = 'D';  cout << "Your grade is " << grade << endl;  }  else  {  grade = 'F';  cout << "Your grade is " << grade << endl;  }    return 0;  } |

* switch 문

- 제어식에 따라 다중 분기 중 하나를 선택하여 실행한다.

- 제어식에 들어 갈 수 있는 값은 정수, 문자, 부울 값, enum 상수이다. 모두 정수에 준하는 값들로 실수가 아니다.



- break와 default는 선택이다. break가 있으면 switch 문을 빠져나가고, 없으면 다음 break 문을 만날 때까지 계속 수행한다.

- 주의: break는 세미콜론(;)으로 끝나고 case와 default는 콜론(:)으로 끝난다.

- switch 문과 다중 if-else 문은 서로 대치할 수 있다.

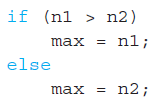
예제: switch 문 – ex02\_02.cpp의 다중 if-else 문을 switch로

ex02\_03.cpp

|  |
| --- |
| #include <iostream>  using namespace std;  int main()  {  int score;  char grade;  cout << "Enter your score: ";  cin >> score;  switch (score / 10)  {  case 9:  grade = 'A';  cout << "Your grade is " << grade << endl;  break;  case 8:  grade = 'B';  cout << "Your grade is " << grade << endl;  break;  case 7:  grade = 'C';  cout << "Your grade is " << grade << endl;  break;  case 6:  grade = 'D';  cout << "Your grade is " << grade << endl;  break;  default:  grade = 'F';  cout << "Your grade is " << grade << endl;  break;  }  return 0;  } |

- 나열형은 정수형 상수리스트며 추후 다시 설명하며, 조건 연산자(3항 연산자)는 사용을 자제한다. 오래된 형식이다.

 ↓

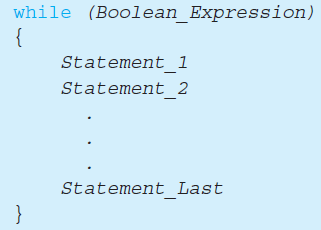
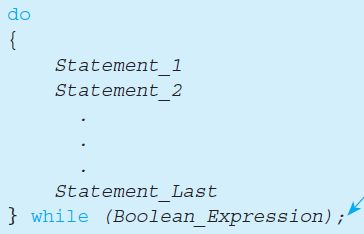


2.3 순환문

* while 문과 do-while 문

- while 문은 한 번도 실행이 안 될 수도 있지만, do-while 문은 반드시 한번은 실행된다.

- do-while 문 끝에는 반드시 세미콜론이 필요하다.

예제: while 문과 do-while 문 – ex02\_03.cpp를 수정

ex02\_04.cpp

|  |
| --- |
| #include <iostream>  using namespace std;  int main()  {  int score, index;  char grade;  cout << "How many people do you have to evaluate? ";  cin >> index;  //while (index > 0)  do  {  cout << "Enter your score: ";  cin >> score;  switch (score / 10)  {  case 9:  grade = 'A';  cout << "Your grade is " << grade << endl;  break;  case 8:  grade = 'B';  cout << "Your grade is " << grade << endl;  break;  case 7:  grade = 'C';  cout << "Your grade is " << grade << endl;  break;  case 6:  grade = 'D';  cout << "Your grade is " << grade << endl;  break;  default:  grade = 'F';  cout << "Your grade is " << grade << endl;  break;  }  index--;  } while (index > 0);  return 0;  } |

* for 문

- for 문은 부울 식을 먼저 평가한 뒤 실행되므로 한 번도 실행되지 않을 수도 있다.

- while 문으로 치환할 수 있다.



- for 문 뒤에는 세미콜론이 오지 않는다. 복합문 사용시에는 중괄호{}를 사용한다.



- 콤마를 이용하여 첫 번째와 마지막 식에 명령어를 추가할 수 있다.



* break 문과 continue 문

- break 문은 루프의 반복을 종료시키며, continue 문은 현재의 반복, 즉 한 번의 루프만 종료시킨다.

- break 문과 continue 문은 되도록이면 사용을 피한다.

예제: for 문, break 문과 continue 문 – ex02\_04.cpp를 수정

ex02\_05.cpp

|  |
| --- |
| #include <iostream>  using namespace std;  int main()  {  int score, index;  char grade;  cout << "How many people do you have to evaluate? ";  cin >> index;  for (int i = 0; i<index; i++)  {  cout << "Enter your score: ";  cin >> score;  //if (score > 100) break;  //if (score > 100) continue;  switch (score / 10)  {  case 9:  grade = 'A';  cout << "Your grade is " << grade << endl;  break;  case 8:  grade = 'B';  cout << "Your grade is " << grade << endl;  break;  case 7:  grade = 'C';  cout << "Your grade is " << grade << endl;  break;  case 6:  grade = 'D';  cout << "Your grade is " << grade << endl;  break;  default:  grade = 'F';  cout << "Your grade is " << grade << endl;  break;  }  }  return 0;  } |

2.4 파일 입력 소개

* ifstream을 이용하여 텍스트 파일(점수) 읽기

- 파일 입력 스트림 클래스

- 읽어 드릴 텍스트 파일을 만들어 솔루션→프로젝트 폴더에 위치시킨다. (경로 지정 가능)

- 파일 입출력을 위해 #include <fstream>가 필요하다.

- ifstream의 객체를 생성한다.

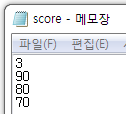
예: ifstream scfile;

- 객체명.open(“파일명”) 으로 텍스트 파일을 연다. 그리고 사용 후에는 객체명.close() 으로 파일을 닫는다.

- 텍스트 파일의 데이터를 읽어 오기 위하여 객체명 >> 변수명; 을 사용한다.

예제: ifstream 객체 이용 파일 입출력 – ex02\_05.cpp를 수정

- score.txt 파일 생성



ex02\_06.cpp

|  |
| --- |
| **#include <iostream>**  **#include <fstream> // 파일 입출력**  **using namespace std;**  **int main()**  **{**  **int score, index;**  **char grade;**  **ifstream scfile;**  **scfile.open("score.txt");**  **scfile >> index;**  **cout << "There are " << index << " people to evaluate." << endl;**    **for (int i = 0; i<index; i++)**  **{**  **scfile >> score;**  **switch (score / 10)**  **{**  **case 9:**  **grade = 'A';**  **cout << "Your grade is " << grade << endl;**  **break;**  **case 8:**  **grade = 'B';**  **cout << "Your grade is " << grade << endl;**  **break;**  **case 7:**  **grade = 'C';**  **cout << "Your grade is " << grade << endl;**  **break;**  **case 6:**  **grade = 'D';**  **cout << "Your grade is " << grade << endl;**  **break;**  **default:**  **grade = 'F';**  **cout << "Your grade is " << grade << endl;**  **break;**  **}**  **} scfile.close();**  **return 0;**  **}** |

* 이름과 점수 읽기

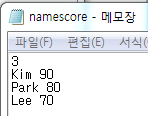
- 이름 즉, 문자열을 처리하기 위해 #include <string> 추가한다.

- 이름을 입력할 변수를 선언한다.

- 이름을 입력하는 부분과 출력하는 부분을 추가하며 나머지 처리 과정은 ex\_06.cpp와 동일하다.

예제: ifstream 객체 이용 파일 입출력 – ex02\_06.cpp를 수정

- namescore.txt 파일 생성



ex02\_07.cpp

|  |
| --- |
| **#include <iostream>**  **#include <fstream> // 파일 입출력**  **#include <string> // 문자 처리**  **using namespace std;**  **int main()**  **{**  **int score, index;**  **char grade;**  **ifstream scfile;**  **string name;**  **scfile.open("namescore.txt");**  **scfile >> index;**  **cout << "There are " << index << " people to evaluate." << endl;**    **for (int i = 0; i<index; i++)**  **{**  **scfile >> name;**  **scfile >> score;**  **switch (score / 10)**  **{**  **case 9:**  **cout << name;**  **grade = 'A';**  **cout << "'s grade is " << grade << endl;**  **break;**  **case 8:**  **cout << name;**  **grade = 'B';**  **cout << "'s grade is " << grade << endl;**  **break;**  **case 7:**  **cout << name;**  **grade = 'C';**  **cout << "'s grade is " << grade << endl;**  **break;**  **case 6:**  **cout << name;**  **grade = 'D';**  **cout << "'s grade is " << grade << endl;**  **break;**  **default:**  **cout << name;**  **grade = 'F';**  **cout << "'s grade is " << grade << endl;**  **break;**  **}**  **} scfile.close();**  **return 0;**  **}** |