Оператори за управление в езика C/C++

Базово програмиране
Лекция 4
Милена Карова
кат. КНТ ТУ-Варна

Логически (булеви) изрази

- Всеки израз, чиято стойност е true или false може да бъде логически. Възможно е операндите да са от всякакъв тип:

5>6 false, ‘A’<‘B’ true

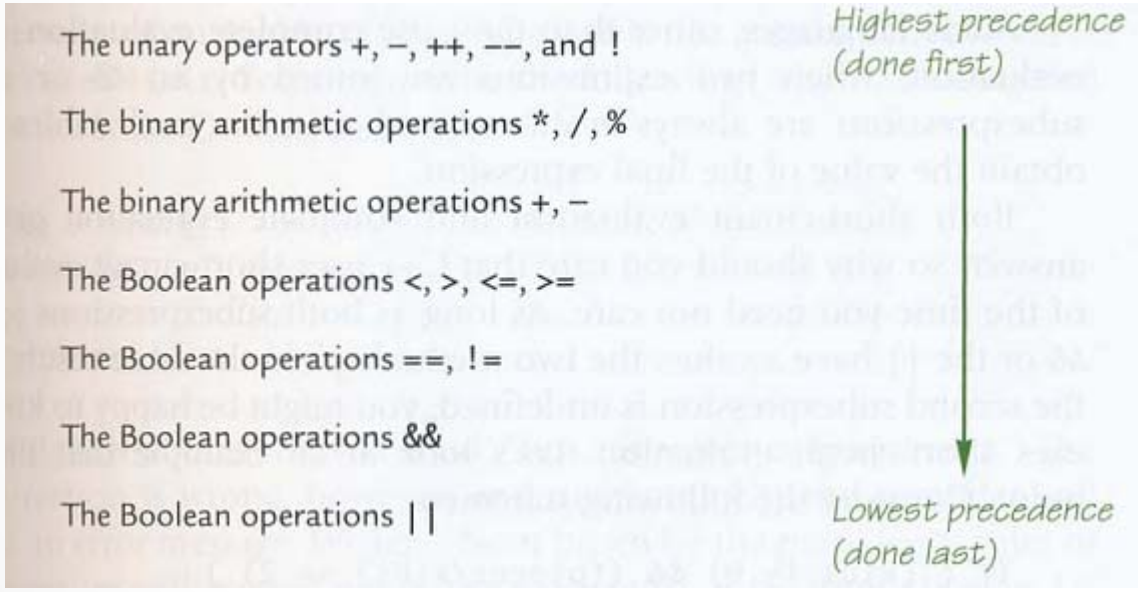
- Използват се за управление на операторите if..then..else, do..while, while…
- В C++ е дефиниран тип bool с възможни стойности на данните: true и false.
- Логически отношения >,<,>=,<=,==,!=
- Логически операции && (and), ||(or), ! (not),^(xor)
- Таблици за истинност

Логически изрази

```
!( ( y < 3) || (y > 7) )
```

```
!( false || true )
```

- Приоритетът на действията е : скоби, not, <,>, &&, ||



- `(x >= 0) && (y > 1)` в този израз в C++ се изчислява само първия операнд

Short circuit evaluation

- ако x= -8 този израз в C++ се изчислява само първия операнд, за общата стойност стойността на втория операнд няма значение (винаги е лъжа). Операцията се нарича short circuit evaluation `(x >= 0) && (y > 1)`
- Ако x=8, следващият израз не се нуждае от изчисляване на втория операнд. Крайната стойност е винаги true (x>=0) | | (y>1)
- В този израз не изчисляването на втория операнд спестява грешката деление на нула ако kids е 0.

```
if ( (kids != 0) && ((pieces/kids) >= 2) )
     cout << "Each child may have two pieces!";
```

Конвертиране на тип int

- C++ използва стойности тип int като булеви стойности
- Всяка ненулева стойност може да се разглежда като true и всяка нулева – като false.
- Логическите изрази ако са изпълнени получават булева стойност true, в противен случай - false

Оператор if

- 3 варианта на if:
  - Прост if

If(time>limit) Something; или If(time>limit)
Something; //няма действие при лъжа

- If..else

If(time>limit)
Something;
else
Somethingelse; //при истина се изпълнява something при лъжа – something else

  - If ..else…if…. вложен if

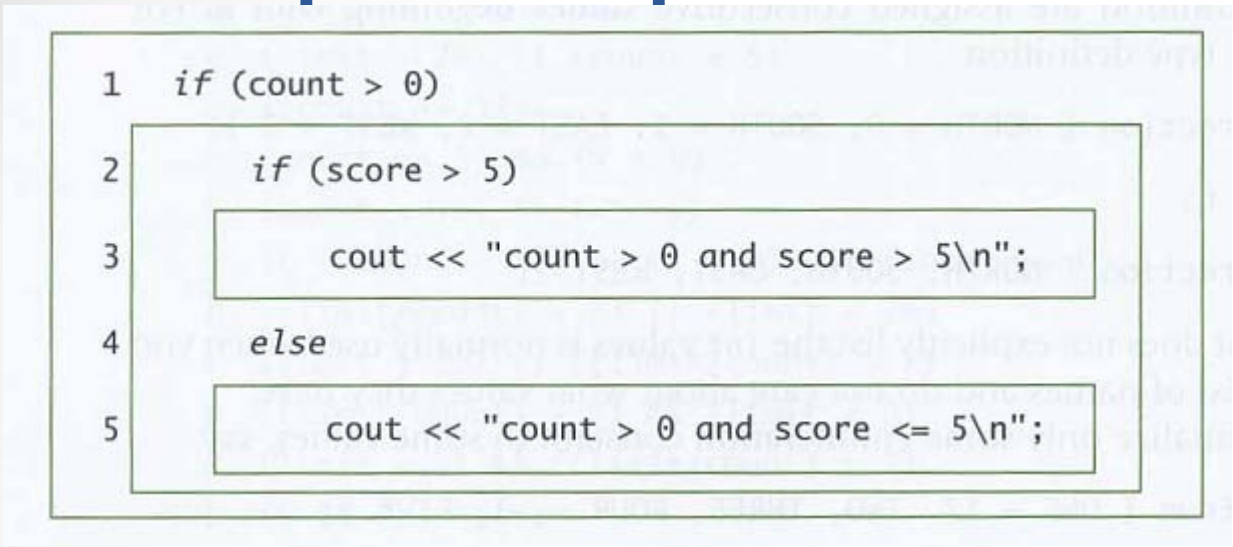
If (time>limit)
Something;
else if (time==limit) something0;
else
Somethingelse;

```
if (!time > limit)
    Something
else
    Something_Else
```

```
(!time) > limit
0 > limit
```

```
if (!(time > limit))
    Something
else
    Something_Else
if (time <= limit)
    Something
else
    Something_Else
```

Пример за вложен if



псевдокод

```
Output "Too high." when guess > number.
Output "Too low." when guess < number.
Output "Correct!" when guess == number.
```

```
if (guess > number)
    cout << "Too high.";
else
    if (guess < number)
        cout << "Too low.";
    else
        if (guess == number)
            cout << "Correct!";
```

```
if (guess > number)
    cout << "Too high.";
else if (guess < number)
    cout << "Too low.";
else if (guess == number)
    cout << "Correct!";
```

```
if ((temperature < -10) && (day == SUNDAY))
    cout << "Stay home.";
else if (temperature < -10) //and day != SUNDAY
    cout << "Stay home, but call work.";
else if (temperature <= 0) //and temperature >= -10
    cout << "Dress warm.";
else //temperature > 0
    cout << "Work hard and play hard.";
```

Съкратен вариант на if

```
if (hours > 40)
    gross_pay = rate*40 + 1.5*rate*(hours - 40);
else
    gross_pay = rate*hours;
```

```
Enter the hourly rate of pay: $20.00
Enter the number of hours worked,
rounded to a whole number of hours: 30
Hours = 30
Hourly pay rate = $20.00
Gross pay = $600.00
```

```
Enter the hourly rate of pay: $10.00
Enter the number of hours worked,
rounded to a whole number of hours: 41
Hours = 41
Hourly pay rate = $10.00
Gross pay = $415.00
```

gross_pay=(hours>40)?rate*40+1.5*rate*(hours-40):rate*hours

```
#include <iostream>
using namespace std;
int main( )
{
    int hours;
    double gross_pay, rate;

    cout << "Enter the hourly rate of pay: $";
    cin >> rate;
    cout << "Enter the number of hours worked,\n"
         << "rounded to a whole number of hours: "
    cin >> hours;

    if (hours > 40)
        gross_pay = rate*40 + 1.5*rate*(hours - 40
    else
        gross_pay = rate*hours;

    cout.setf(ios::fixed);
    cout.setf(ios::showpoint);
    cout.precision(2);
    cout << "Hours = " << hours << endl;
    cout << "Hourly pay rate = $" << rate << endl;
    cout << "Gross pay = $" << gross_pay << endl;

    return 0;
}
```

Мултиразклонение

```
//Program to illustrate the switch statement.
#include <iostream>
using namespace std;

int main( )
{
    char grade;

    cout << "Enter your midterm grade and press Return: ";
    cin >> grade;

    switch (grade)
    {
        case 'A':
            cout << "Excellent. "
                 << "You need not take the final.\n";
            break;
        case 'B':
            cout << "Very good. ";
            grade = 'A';
            cout << "Your midterm grade is now "
                 << grade << endl;
            break;
        case 'C':
            cout << "Passing.\n";
            break;
        case 'D':
        case 'F':
            cout << "Not good. "
                 << "Go study.\n";
            break;
        default:
            cout << "That is not a possible grade.\n";
    }

    cout << "End of program.\n";
    return 0;
}
```

```
Enter your midterm grade and press Return: A
Excellent. You need not take the final.
End of program.
```

```
Enter your midterm grade and press Return: B
Very good. Your midterm grade is now A.
End of program.
```

```
Enter your midterm grade and press Return: D
Not good. Go study.
```

```
Enter your midterm grade and press Return: E
That is not a possible grade.
End of program.
```

Селекторът на switch може да бъде тип int, char, bool, enum дефиниция

Характеристики на switch

- Селектор
- Оператор break
- Повече от една стойност след case

```
case 'A':
case 'a':
    cout << "Excellent. "
         << "You need not take the final.\n";
    break;
```

- Default стойност

```
int vehicle_class;
cout << "Enter vehicle class: ";
cin >> vehicle_class;

switch (vehicle_class)
{
    case 1:
        cout << "Passenger car.";
        toll = 0.50;
        break;
    case 2:
        cout << "Bus.";
        toll = 1.50;
        break;
    case 3:
        cout << "Truck.";
        toll = 2.00;
        break;
    default:
        cout << "Unknown vehicle class!";
}
```

If you forget this break, then passenger cars will pay $1.50.

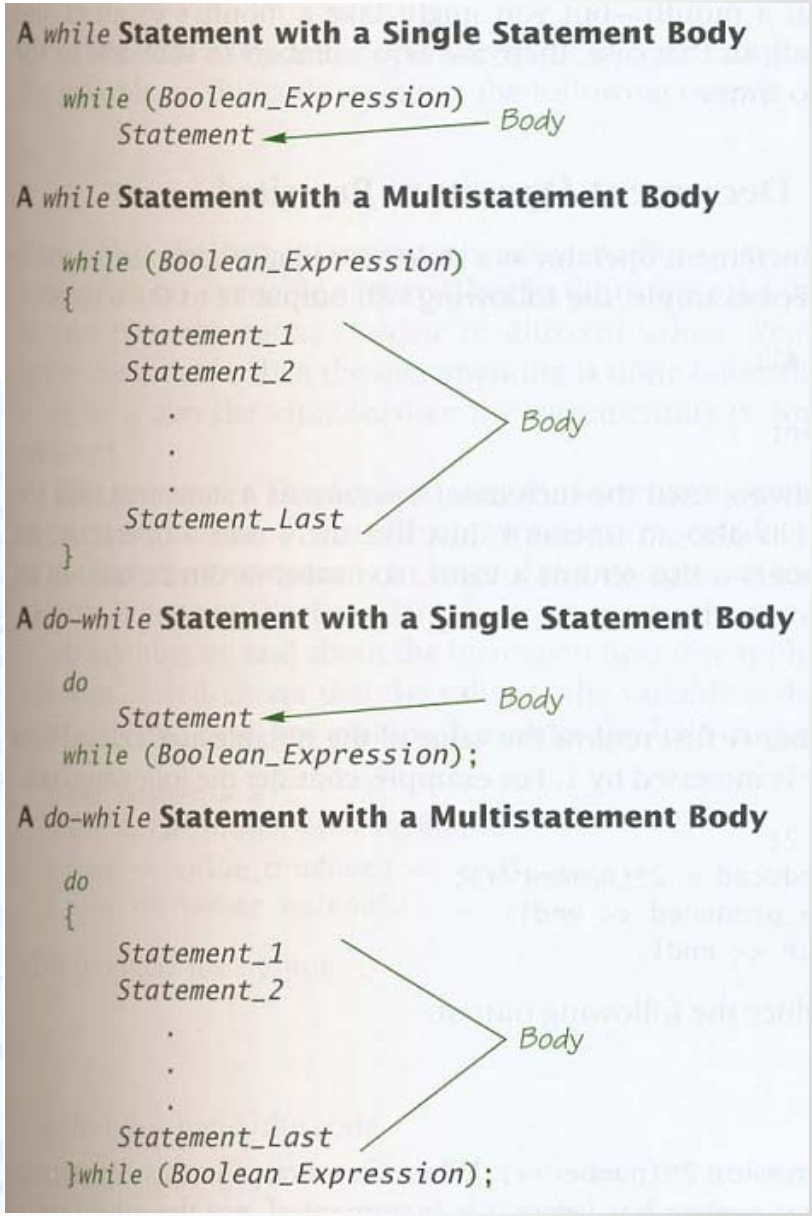
Използване на switch оператор за меню

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


int main( )
{
    int choice;

    do
    {
        cout << endl
             << "Choose 1 to see the next homework assignment.\n"
             << "Choose 2 for your grade on the last assignment.\n"
             << "Choose 3 for assignment hints.\n"
             << "Choose 4 to exit this program.\n"
             << "Enter your choice and press Return: ";
        cin >> choice;

        switch (choice)
        {
             case 1:
                //code to display the next assignment on screen would go here.
                 break;
             case 2:
                //code to ask for a student number and give the corresponding
                //grade would go here.
                 break;
             case 3:
                //code to display a hint for the current assignment would go
                //here.
                 break;
             case 4:
                 cout << "End of Program.\n";
                 break;
             default:
                  cout << "Not a valid choice.\n"
                       << "Choose again.\n";
        }
    }while (choice != 4);

    return 0;
```

Цикъл

- Определение за цикъл (loop)
- Тяло на цикъла (loop body)
- Преминаване (loop iteration)
- Разлика между while и do..while
- Брояч на цикъла

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

int main( )
{
    int number_of_items, count,
        calories_for_item, total_calories;

    cout << "How many items did you eat today? ";
    cin >> number_of_items;

    total_calories = 0;
    count = 1;
    cout << "Enter the number of calories in each of the\n"
         << number_of_items << " items eaten:\n";

    while (count++ <= number_of_items)
    {
        cin >> calories_for_item;
        total_calories = total_calories
                       + calories_for_item;
    }

    cout << "Total calories eaten today = "
         << total_calories << endl;
    return 0;
}
```



Оператор за цикъл for

- Използване с подредени данни

```
sum = 0;
n = 1;
while (n <= 10)
{
    sum = sum + n;
    n++;
}
```

```
sum = 0;
for (n = 1; n <= 10; n++)
    sum = sum + n;
```

```
n = 1; n <= 10; n++ and sum = sum + n;
```

- Използване на декларация на променливи в скобите на for

SYNTAX

```
for (Initialization_Action; Boolean_Expression; Update_Action)
    Body_Statement
```

EXAMPLE

```
for (number = 100; number >= 0; number--)
    cout << number
         << " bottles of beer on the shelf.\n";
```

```
Initialization_Action;
while (Boolean_Expression)
{
    Body_Statement
    Update_Action;
}
```

EQUIVALENT EXAMPLE

```
number = 100;
while (number >= 0)
{
    cout << number
         << " bottles of beer on the shelf.\n";
    number--;
}
```

For оператор

- Проблеми

```
for (int count = 1; count <= 10; count++);
    cout << "Hello\n";
```

Problem semicolon

```
for (int count = 1; count <= 10; count++);
```

- Какво прави тази програма

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

int main()
{
    int number, sum = 0, count = 0;
    cout << "Enter 10 negative numbers:\n";

    while (++count <= 10)
    {
        cin >> number;

        if (number >= 0)
        {
            cout << "ERROR: positive number"
                 << " or zero was entered as the\n"
                 << count << "th number! Input ends "
                 << "with the " << count << "th number.\n"
                 << count << "th number was not added in.\n";
            break;
        }

        sum = sum + number;
    }

    cout << sum << " is the sum of the first "
         << (count - 1) << " numbers.\n";

    return 0;
}
```

```
Enter 10 negative numbers:
-1 -2 -3 4 -5 -6 -7 -8 -9 -10
ERROR: positive number or zero was entered as the
4th number! Input ends with the 4th number.
4th number was not added in.
-6 is the sum of the first 3 numbers.
```

Безкраен цикъл

```
x = 2;
while (x != 12)
{
    cout << x << endl;
    x = x + 2;
}
```

```
for (x=12;x>=2;x++)
 { cout<<x<<endl;}
```

- while (true)

{……..

  ……..}

- for(;;)

{……..

  ……..}

Проектиране на цикли

- Намиране на суми и произведения (начална стойност на сумата 0, на произведението 1
- Цикъл с определен брой повтарящи се итерации
- Цикъл, в който се пита дали да се повтори тялото на цикъла
- Цикъл, който се повтаря докато се въвеждат данни (от клавиатура, от файл)
- Цикъл, който се повтаря при определено условие (flag condition) compute_grade в примера е flag

```
int n = 1;
grade = compute_grade(n);
while (grade < 90)
{
    n++;
    grade = compute_grade(n);
}
cout << "Student number " << n << " may be a tutor.\n"
     << "This student has a score of " << grade << endl;
```

Вложени цикли -пример

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

int main()
{
cout << "This program tallies conservationist reports\n"
          << "on the green-necked vulture.\n"
          << "Each conservationist's report consists of\n"
          << "a list of numbers. Each number is the count of\n"
          << "the eggs observed in one"
          << "green-necked vulture nest.\n"
          << "This program then tallies
          << "the total number of eggs.\n";

    int number_of_reports;
    cout << "How many conservationist reports are there? ";
    cin >> number_of_reports;

    int grand_total = 0, subtotal, count;
    for (count = 1; count <= number_of_reports; count++)
    {
        cout << endl << "Enter the report of "
             << "conservationist number " << count << endl;
        cout << "Enter the number of eggs in each nest.\n"
             << "Place a negative integer at the end of your list.\n";"
        subtotal = 0;
        int next;
        cin >> next;
        while (next >=0)
        {
            subtotal = subtotal + next;
            cin >> next;
        }
        cout << "Total egg count for conservationist "
             << " number " << count << " is "
             << subtotal << endl;
        grand_total = grand_total + subtotal;
    }
    cout << endl << "Total egg count for all reports = "
         << grand_total << endl;

    return 0;
}
```