An Introduction to Programming though C++

Abhiram G. Ranade

Lecture Sequence 3.1

Ch. 6: Conditional Execution

Let us calculate income tax

Write a program to read income and print income tax, using following rules.

- If income ≤ 180,000, then tax = 0.
- If 180,000 ≤ income ≤ 500,000, then tax = 10% of (income - 180,000).
- If 500,000 ≤ income ≤ 800,000, then tax = 32,000 + 20% of (income – 500,000).
- If income > 800,000, then tax = 92,000 + 30% of (income – 800,000).

Cannot write tax calculation program using what you have learnt so far.

Outline

- Basic If statement
  - Program to solve a very simple tax problem
- If-else statement
  - Better program to solve the simple problem
- Most general if statement form
  - Full tax calculation program
- How to express complex conditions
- Case study: A different way to control the turtle
- The switch statement
  - Yet another way to control the turtle
- Logical data

Basic if statement

- Form:

if (condition) consequent

- ***condition***: “boolean” expression.
- “boolean” : Should evaluate to “true” or “false”.
- ***consequent***: C++ statement, e.g. assignment.
- ***consequent*** could also be a block, i.e. {...}
- If ***condition*** evaluates to true, then the ***consequent*** is executed.
- If ***condition*** evaluates to false, then ***consequent*** is ignored.

Conditions

Simple condition: ***exp1 relop exp2***

- ***relop*** : relational operator:

< : less than. <= : less than or equal. == : equal.

> : greater than. >= : greater than or equal. != : not equal

- Condition is considered true if ***exp1*** relates to ***exp2*** as per the specified relational operator ***relop***.

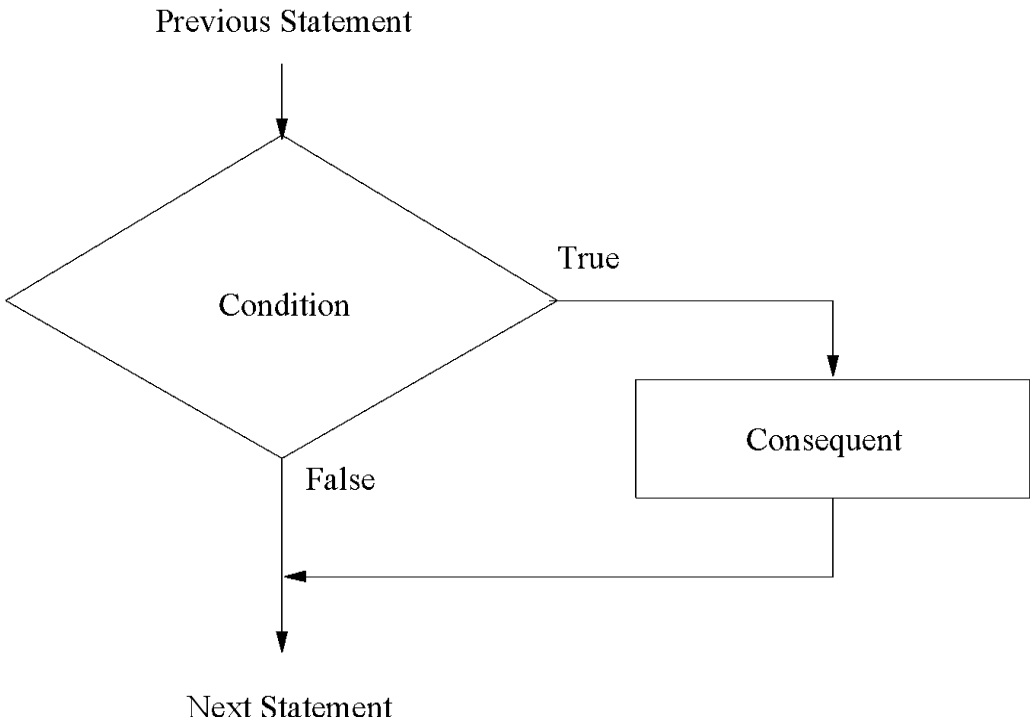
Suppose ***x = 5, y = 10, z = 100***.

- ***x >= y*** is false.
- ***x*x > y*** is true.
- ***x*y – z == 10*** is false

Flowchart

- Pictorial representation of a program.
- Statements put inside boxes.
- If box C will possibly be executed after box B, then put an arrow from B to C.
- Specially convenient for showing conditional execution, because there can be more than one “next” statements.
- “Diamond” shaped boxes are used for condition checks.

Flowchart of ***if(condition) consequent***



Simplified problem: just determine if any tax is owed

```
main_program{
 float income, tax; cin >> income;
 if(income <= 180000)
      cout << "No tax owed." << endl;
 if(income > 180000)
   cout << "You owe tax." << endl;
}
// Always checks both conditions.
// If the first condition is true,
// then you know second must be false,
// and vice versa.
// Can we avoid checking twice?
```

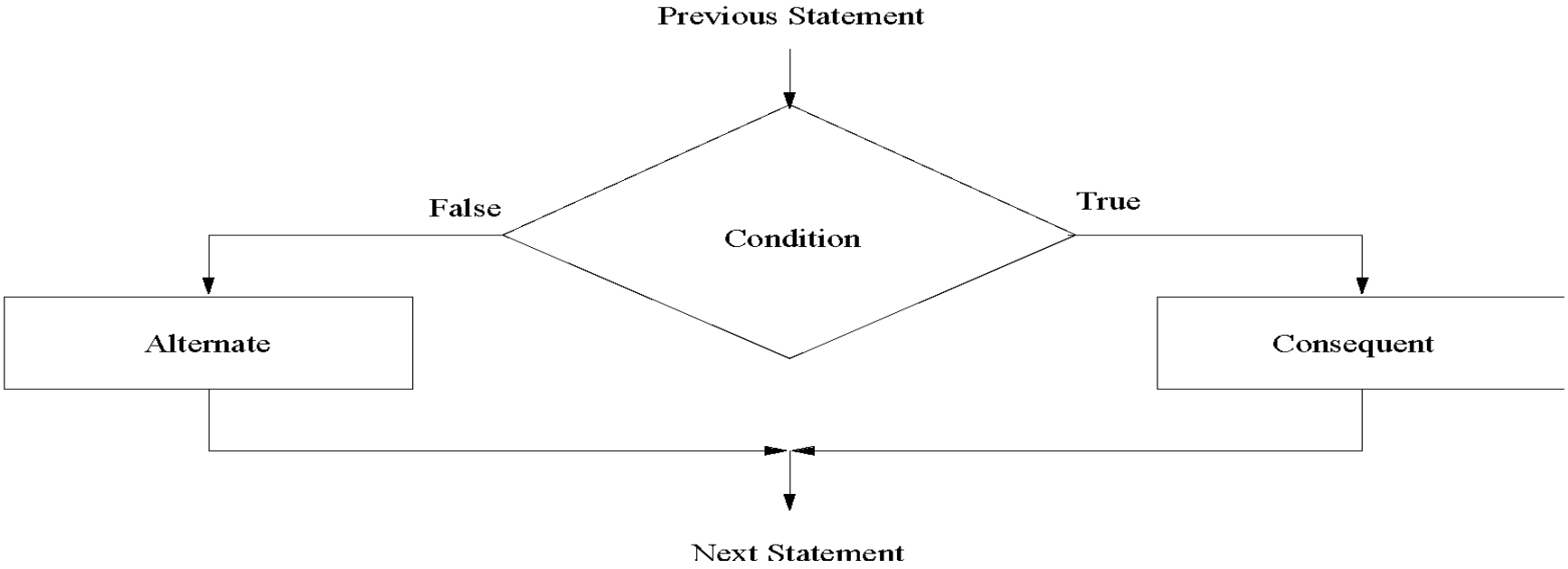
Another form of if

if (condition) consequent
else alternate

- The ***condition*** is first evaluated.
- If it is true, then ***consequent*** is executed.
- If ***condition*** is false, then ***alternate*** is executed.

alternate can also be a block.

If else flowchart



Better program for simple problem

```
main_program{
    float income, tax; cin >> income;
    if(income <= 180000)
        cout << "No tax owed." << endl;
    else
        cout << "You owe tax." << endl;
}
// Only one condition check. Thus
// more efficient than previous.
```

Exercise

- Write a program that reads in a number and prints its square root. If the number is positive, it should use the sqrt function. If the number is negative, it should invoke sqrt on the negative of the number (which will be a positive quantity) and print the result followed by the letter ‘i’, to indicate that the result is imaginary.

What we discussed

- 2 Forms of the if statement.

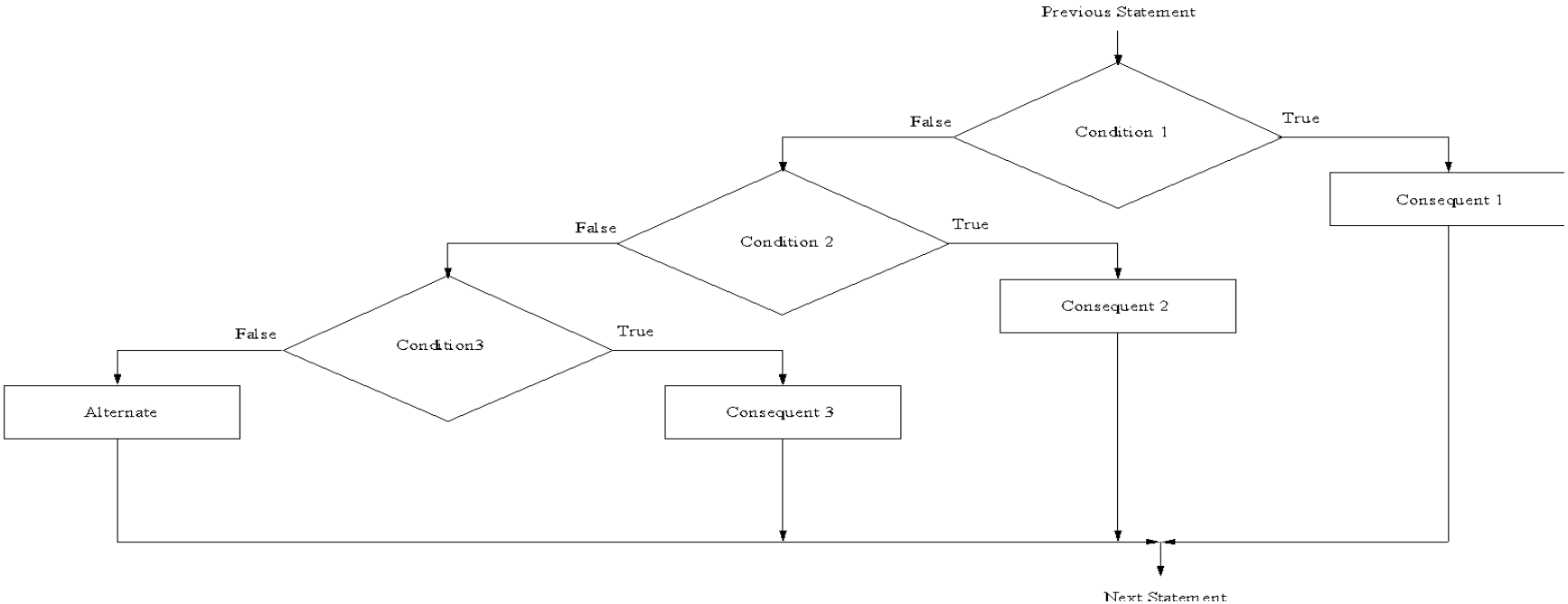
Next: The most general form of the if statement.

Most general form of if

```
if (condition1) consequent1
else if (condition2) consequent2
...
else if (conditionn) consequentn
else alternate      // optional
```

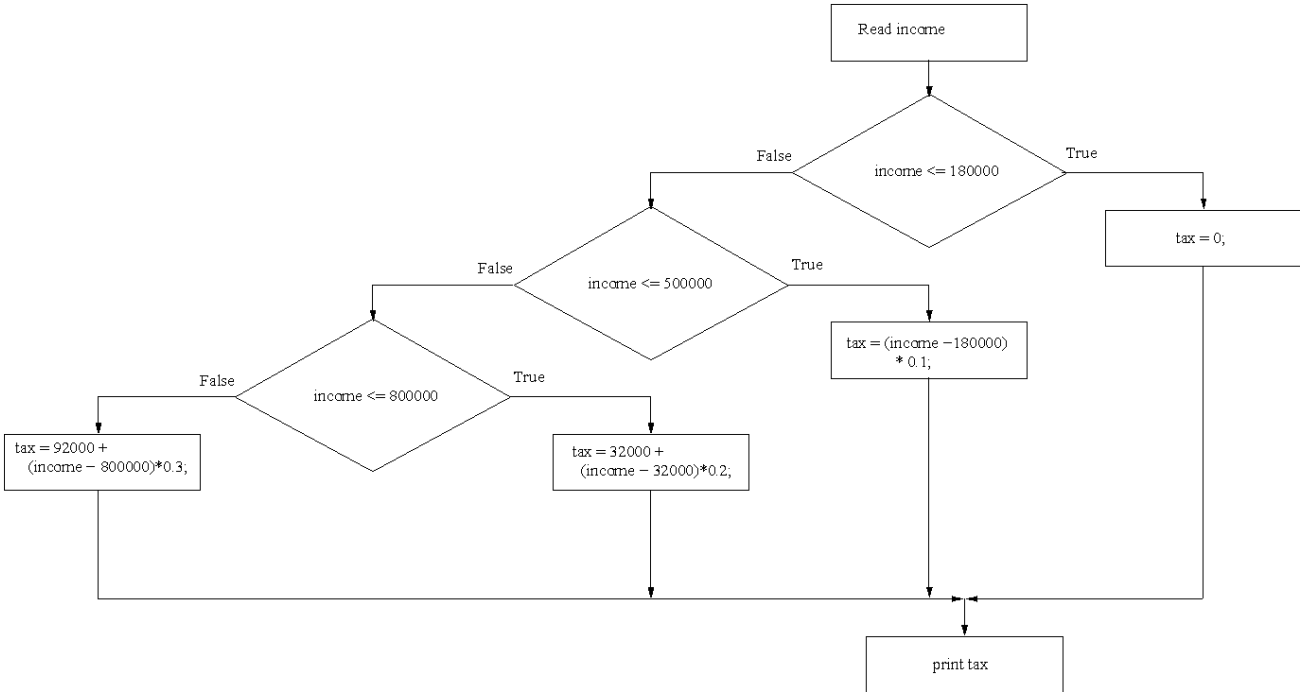
- Evaluate conditions in order.
- Some condition true: execute corresponding consequent. Do not evaluate subsequent conditions.
- All conditions false: execute alternate if specified.
- Consequents and alternate can be blocks or single statements.

General if example flowchart (with 3 conditions)



Tax calculation program

```
main_program{
  float tax,income; cin >> income;
  if (income <= 180000) tax = 0;
  else if(income <= 500000)
       tax = (income — 180000) * 0.1;
  else if(income <= 800000)
       tax = (income — 500000) * 0.2 + 32000;
  else tax = (income — 800000) * 0.3 + 92000;
  cout << tax << endl;
}
```


Read income
income <= 180000
False
True
tax = 0;
income <= 500000
False
True
tax = (income −180000)
* 0.1;
income <= 800000
False
True
tax = 92000 +
(income − 800000)*0.3;
tax = 32000 +
(income − 32000)*0.2;
print tax

Exercise: Is the following program correct? Precisely state the error, if any.

```
main_program{
  float tax,income; cin >> income;
  if (income <= 180000) tax = 0;
  if(income <= 500000)
    tax = (income — 180000) * 0.1;
  if(income <= 800000)
    tax = (income — 500000) * 0.2 + 32000;
  else tax = (income — 800000) * 0.3 + 92000;
  cout << tax << endl;
}
```

What we discussed

- Most general form of if statement
- Use in tax calculation program

Next: more general ways of specifying conditions