

if

Abhiram Ranade

# Let us calculate income tax

- If income  $\leq 1,80,000$ , then tax = 0.
- If income is between 180,000 and 500,000 then tax = 10% of (income - 180,000).
- If income is between 500,000 and 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.**

# Program

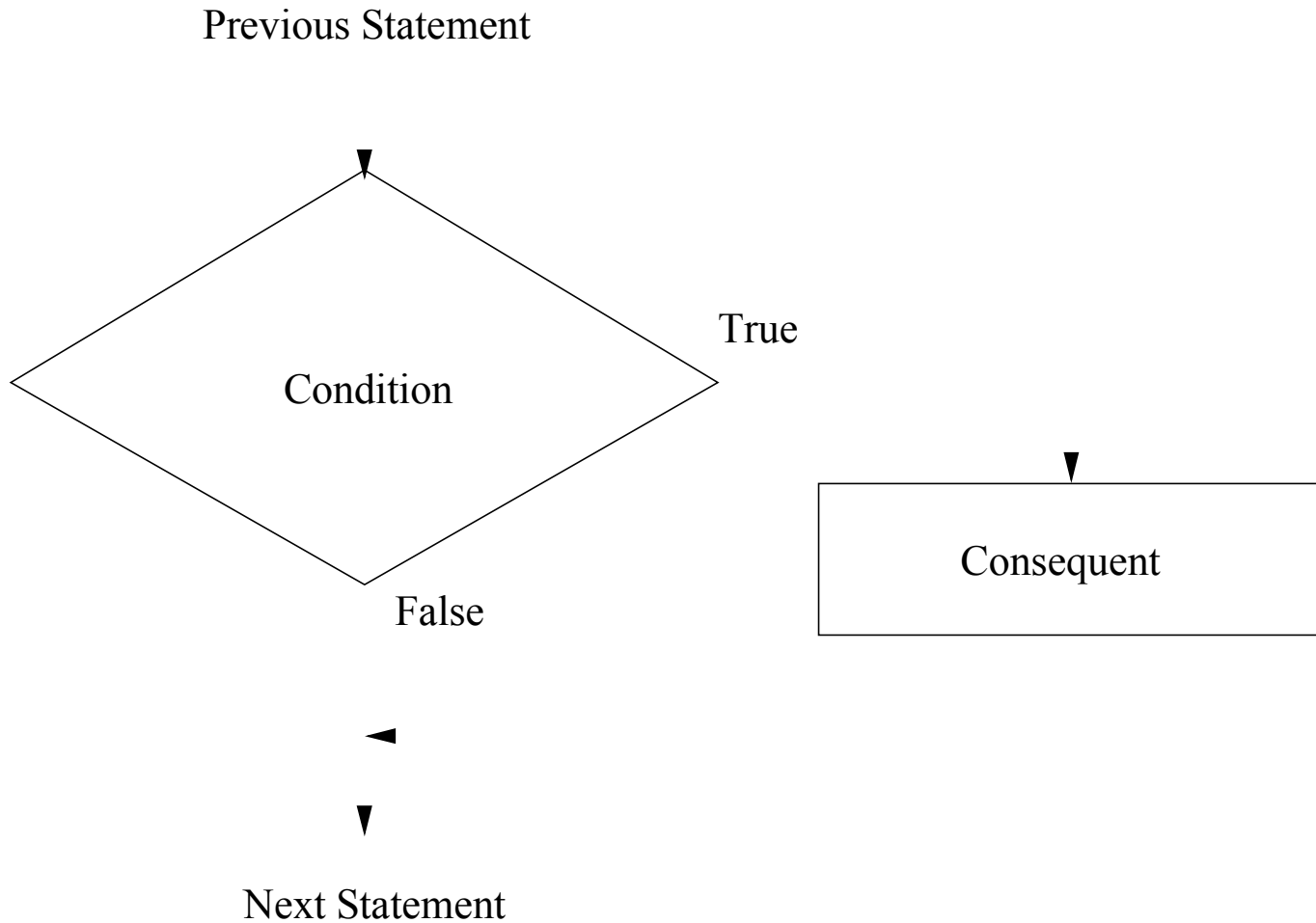
```
main_program{
    float income, tax;    cin >> income;
    if(income <= 180000) tax = 0;
    if((income > 180000) && (income <= 500000))
        tax = (income - 180000)* 0.1;
    if((income > 500000) && (income <= 800000))
        tax = 32000+(income - 500000)* 0.2;
    if(income > 800000)
        tax = 92000+(income - 800000)* 0.3;
    cout << "Tax is: " << tax << endl;
}
```

# The if statement

## if (condition) consequent

- condition: boolean expression. Should evaluate to true or false.
- consequent: C++ statement, e.g. assignment.
- If condition evaluates to true, then the consequent is executed.
- If condition evaluates to false, then consequent is ignored.

# Flowchart of if



# Conditions

- Simple condition =  $\text{exp1 relop exp2}$
- relational operator:  $<$ ,  $<=$ ,  $==$ ,  $>$ ,  $>=$ ,  $!=$
- Operators respectively mean less than, less than or equal, equal, greater than, greater than or equal, not equal.
- Condition is true if  $\text{exp1}$  relates to  $\text{exp2}$  as per the specified relational operator  $\text{relop}$ .

# More complex conditions

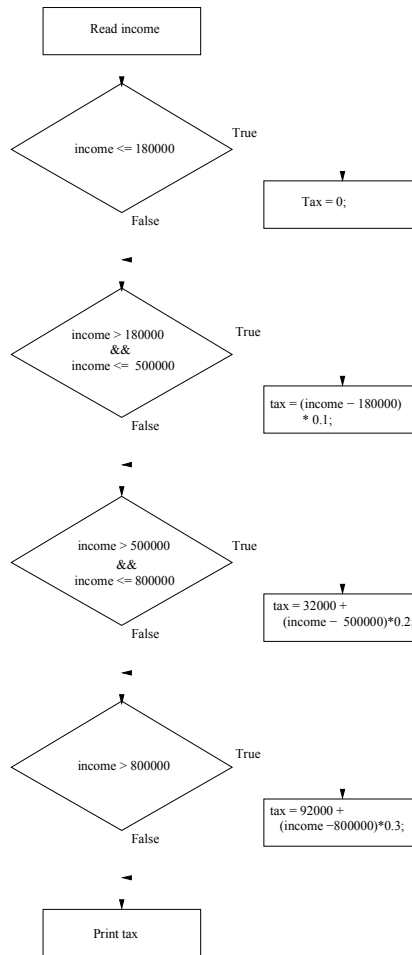
- `condition1 && condition2` : true only if both true. “AND”
- `condition1 || condition2` : true only if at least one is true. “OR”
- `! condition` : true only if `condition` is false.
- Components of complex conditions may themselves be complex conditions.

`!((income < 180000) || (income > 500000))`

# Program Execution

```
main_program{  
    float income, tax;    cin >> income;  
    if(income <= 180000) tax = 0;  
    if((income > 180000) && (income <= 500000))  
        tax = (income - 180000)* 0.1;  
    if((income > 500000) && (income <= 800000))  
        tax = 32000+(income - 500000)* 0.2;  
    if(income > 800000)  
        tax = 92000+(income - 800000)* 0.3;  
    cout << "Tax is: " << tax << endl;  
} // Every condition is checked. Only one can be  
    true.
```





# Remark

- Consequent may be a block containing several statements. If condition is true, all statements in the block are executed, in order.

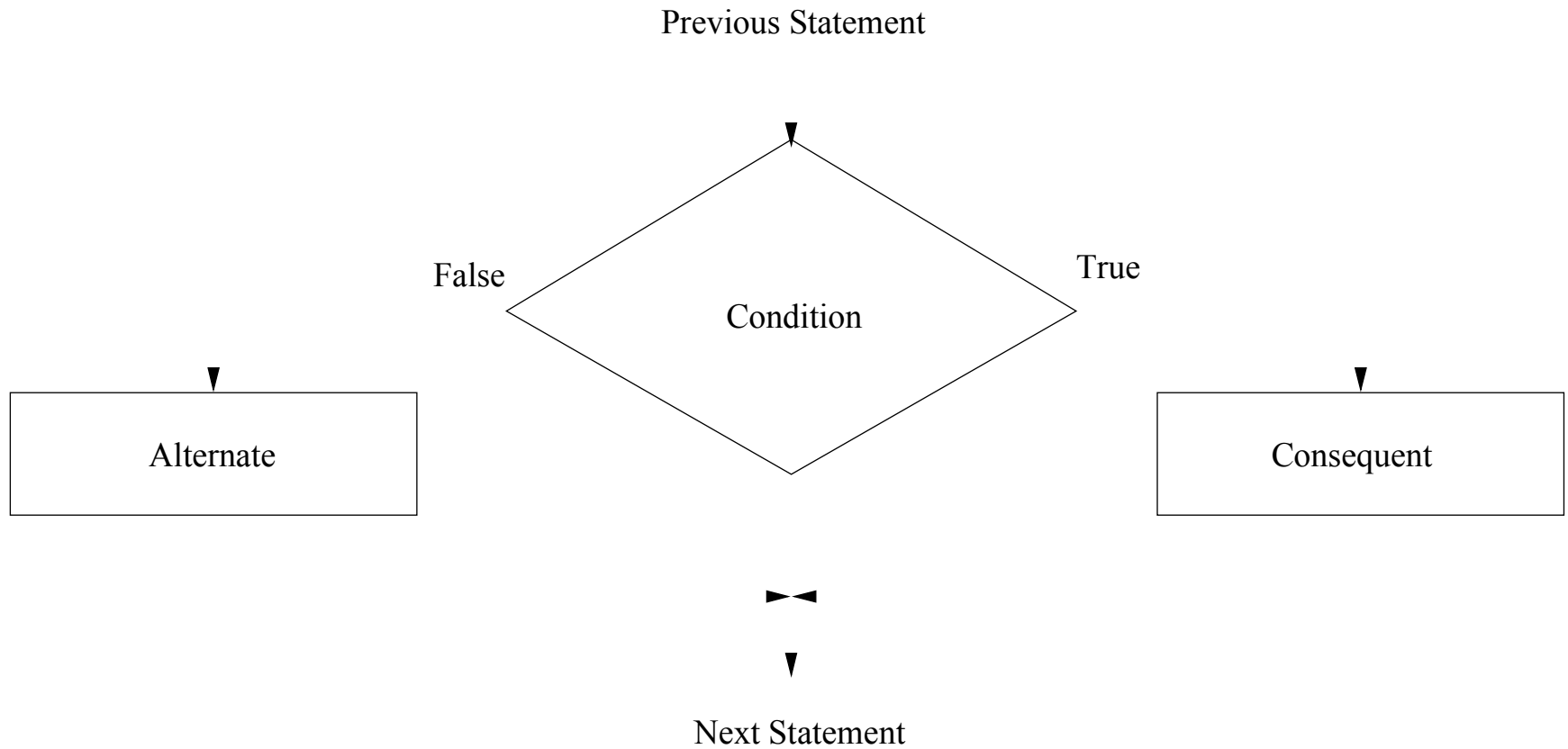
```
if (income > 800000){  
    tax = 92000 + (income - 800000) *  
        0.3;  
    cout << "You are in highest tax  
    bracket.\n";  
}
```

# 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.

# If else flowchart



# Most general form

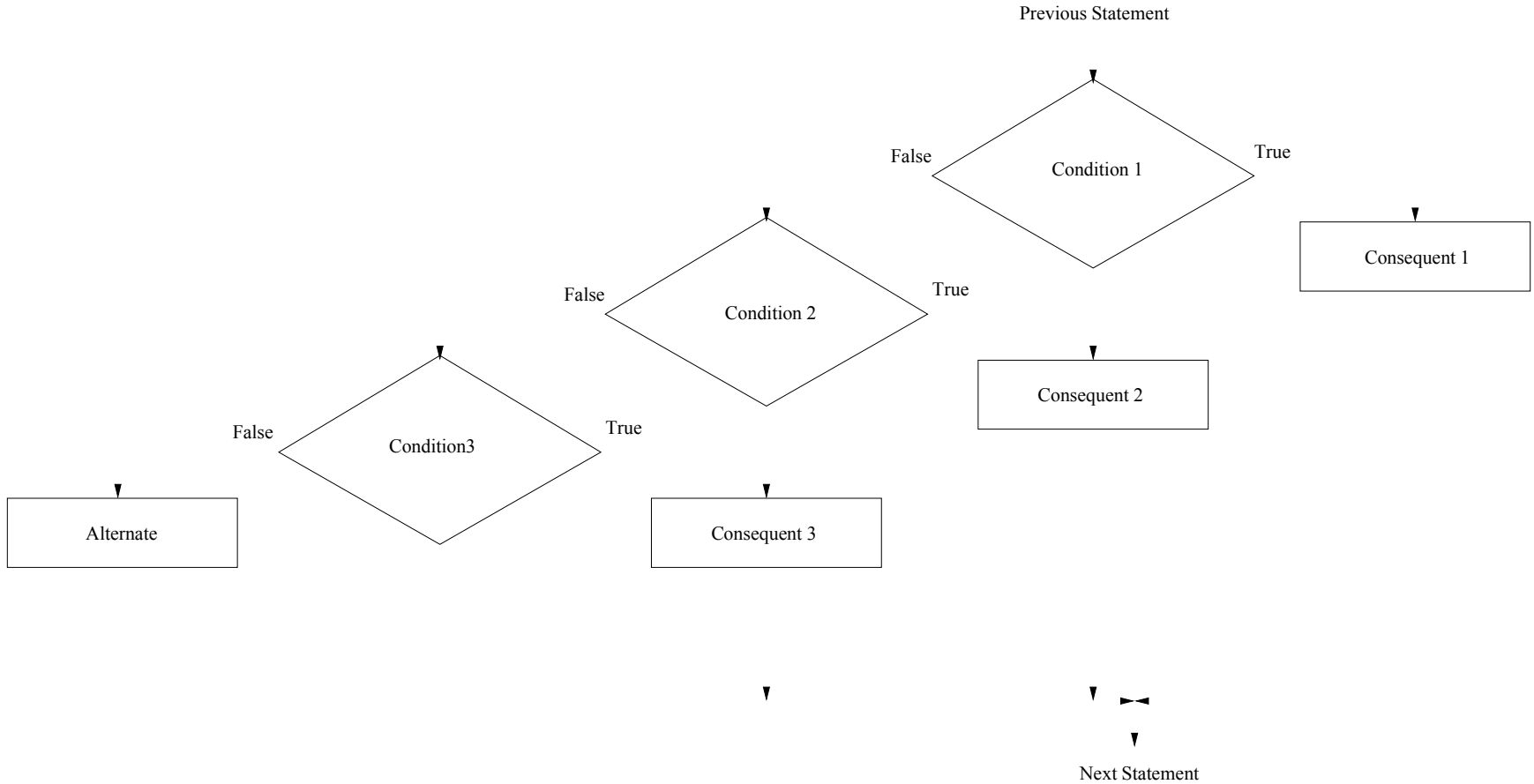
if (condition1) consequent1  
else if (condition2) consequent2

...

else if (conditionn) consequentn  
else alternate

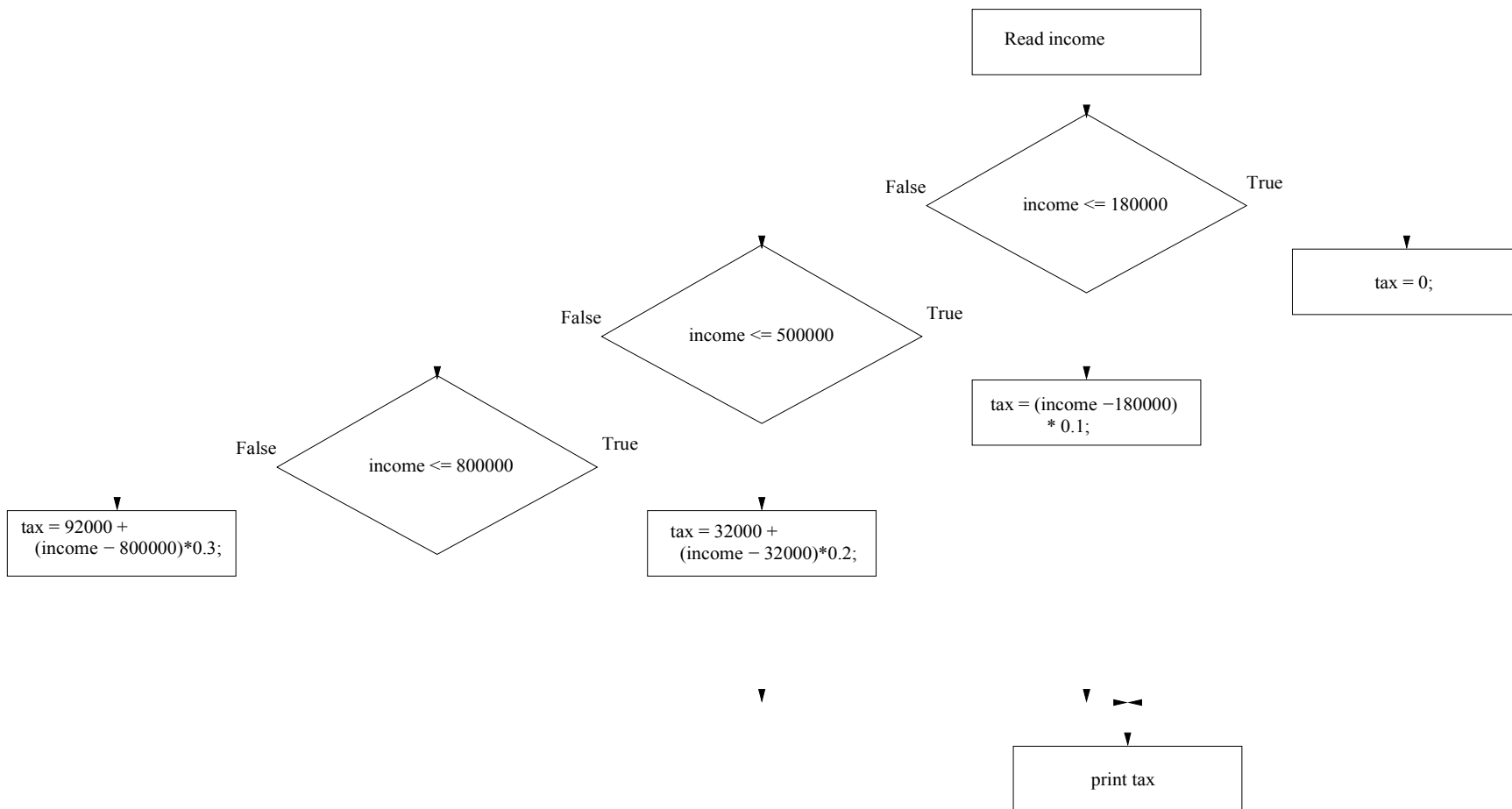
- Evaluate conditions in order.
- Some **condition<sub>i</sub>** true: execute **consequent<sub>i</sub>**. Do not evaluate subsequent conditions.
- All conditions false: execute **alternate**.

# General if example flowchart



# New income tax program

```
main_program{
    float tax,income; cin >> income;
    if (income <= 180000) tax = 0;
    else if(income <= 500000) // enough?
        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;
}
```





# Examples from book

Turtle controller

“Button” based turtle controller.