Programming with C++

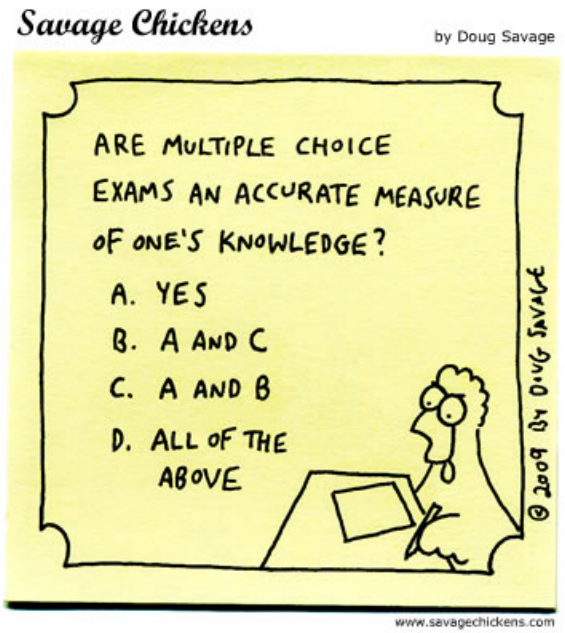
COMP2011: C++ Control II

Cecia Chan
Cindy Li

Department of Computer Science & Engineering
The Hong Kong University of Science and Technology
Hong Kong SAR, China

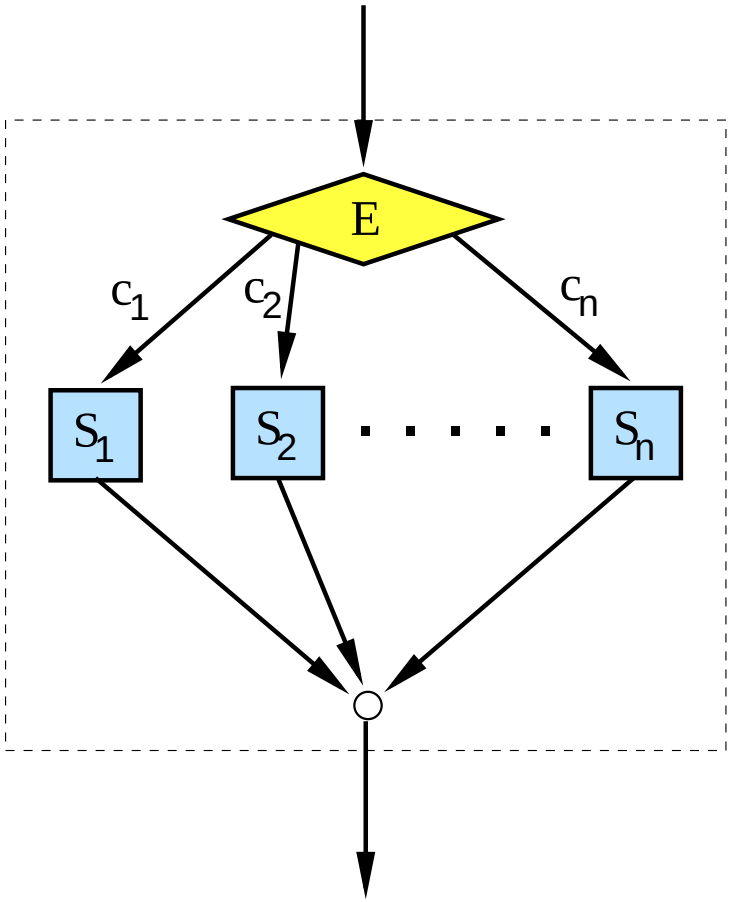
Part I

Let's **switch**: C++ Multiple Choices



switch Statement

switch statement is a variant of the if-else-if statement, that allows multiple choices based on the value of an integral expression.



Syntax: switch Statement

```
switch (integral expression)
{
    case constant-1:
        statement-sequence-1;
        break;
    case constant-2:
        statement-sequence-2;
        break;
    ...
    case constant-N:
        statement-sequence-N;
        break;
    default: // optional
        statement-sequence-(N+1);
}
```

Example: switch on Integers

```
#include <iostream>     /* File: switch-find-comp2011-instructor.cpp */
using namespace std;

int main()              // To determine your instructor
{
    cout << "Enter the COMP2011 section number to find its instructor: ";
    int section;        // COMP2011 section number: should be 1, 2, 3, or 4
    cin >> section;     // Input COMP2011 section number

    switch (section)
    {
        case 1:
            cout << "Sergey Brin" << endl; break;
        case 2:
            cout << "Bill Gates" << endl; break;
        case 3:
            cout << "Steve Jobs" << endl; break;
        case 4:
            cout << "Jeff Bezos" << endl; break;
        default:
            cerr << "Error: Invalid lecture section " << section << endl;
            break;
    }

    return 0;
}
```

Example: switch on Characters

```
#include <iostream>     /* File: switch-char-bloodtype.cpp */
using namespace std;

int main()              // To find out who may give you blood
{
    cout << "Enter your blood type (put 'C' for blood type AB): ";
    char bloodtype; cin >> bloodtype;

    switch (bloodtype)
    {
        case 'A':
            cout << "Your donor must be of blood type: O or A\n";
            break;
        case 'B':
            cout << "Your donor must be of blood type: O or B\n";
            break;
        case 'C':
            cout << "Your donor must be of blood type: O, A, B, or AB\n";
            break;
        case 'O':
            cout << "Your donor must be of blood type: O";
            break;
        default:        // To catch errors
            cerr << "Error: " << bloodtype << " is not a valid blood type!\n";
            break;
    }
    return 0;
}
```

Example: switch with Sharing Cases

```
#include <iostream>      /* File: switch-int-grade.cpp */
using namespace std;

int main()               // To determine your grade (fictitious)
{
    char grade;          // Letter grade
    int mark;            // Numerical mark between 0 and 100
    cin >> mark;

    switch (mark/10)
    {
        case 10:         // Several cases may share the same action
        case 9:
            grade = 'A'; break; // If mark >= 90
        case 8: case 7: case 6: // May write several cases on 1 line
            grade = 'B'; break; // If 90 > mark >= 60
        case 5:
        case 4:
        case 3:
        case 2:
            grade = 'C'; break; // If 60 > mark >= 20
        case 1:
            grade = 'D'; break; // If 20 > mark >= 10
        default:
            grade = 'F'; break;
    }

    cout << "Your letter grade is " << grade << endl;
    return 0;
}
```

Example: switch vs. if-else-if

```
#include <iostream>      /* File: if-elseif-grade.cpp */
using namespace std;

int main()               /* To determine your grade (fictitious) */
{
    char grade;          // Letter grade
    int mark;            // Numerical mark between 0 and 100
    cin >> mark;

    if (mark >= 90)
        grade = 'A';     // mark >= 90

    else if (mark >= 60)
        grade = 'B';     // 90 > mark >= 60

    else if (mark >= 20)
        grade = 'C';     // 60 > mark >= 20

    else if (mark >= 10)
        grade = 'D';     // 20 > mark >= 10

    else
       grade = 'F';      // 10 > mark

    cout << "Your letter grade is " << grade << endl;
    return 0;
}
```

Remarks on switch

- The expression for switch must evaluate to an integral value (integer, char, bool in C++).
- NO 2 cases may have the same value.
- On the other hand, several cases may share the same action statements.
- When a case constant is matched, the statements associated with the case are executed until either
  - a break statement.
  - a return statement.
  - the end of the switch statement.
- Difference between a switch statement and a if-else-if statement:
  - switch statement can only test for equality of the value of one quantity.
  - each expression of the if-else-if statement may test the truth value of different quantities or concepts.

Example: Give me a break

```
#include <iostream>      /* File: switch-no-break.cpp */
using namespace std;

int main()               // To determine your grade (fictitious)
{
    char grade;          // Letter grade
    int mark;            // Numerical mark between 0 and 100
    cin >> mark;

    /* What happens if you forget to break? What is the output? */
    switch (mark/10)
    {
        case 10: case 9:
            cout << "Your grade is A" << endl;
        case 8: case 7: case 6:
            cout << "Your grade is B" << endl;
        case 5: case 4: case 3: case 2:
            cout << "Your grade is C" << endl;
        case 1:
            cout << "Your grade is D" << endl;
        default:
            cout << "Your grade is F" << endl;
    }

    return 0;
}
```

New Data Types with enum

- One way to define a new data type is to use the keyword enum.

Syntax: enum Declaration

enum new-datatype { identifier1 [=value1], identifier2 [=value2], ··· };

Example

```
enum weekday { MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY,
               SATURDAY, SUNDAY };          // 0,1,2,3,4,5,6

enum primary_color { RED = 1, GREEN, BLUE }; // 1,2,3

enum bloodtype { A, B, AB = 10, O };         // 0,1,10,11
```

User-defined enum Type

- An enumeration is a type that can hold a finite set of symbolic objects.
- The symbolic (meaningful) names of these objects follow the same rule as identifier names.
- The symbolic names make your program easier to read/understand.
- Internally, these objects are represented as integers.
- By default, the first object is given the value zero, then each subsequent object is assigned a value one greater than the previous object's value.
- The integral values of the enumerated objects may be assigned other integral values by the programmer.
- Thus, the objects of an enum type act like named integer constants.

Example: enum with switch

```
#include <iostream>     /* File: enum-shapes.cpp */
using namespace std;

int main()
{
    enum shapes { TEXT, LINE, RECT, CIRCLE };
    cout << "supported shapes: "
         << " TEXT = " << TEXT << " LINE = " << LINE
         << " RECT = " << RECT << " CIRCLE = " << CIRCLE << endl;
    int myshape; // Why the type of myshape is not shape?
    cin >> myshape;

    switch (myshape)
    {
        case TEXT:
            cout << "Call a function to print text" << endl; break;
        case LINE:
            cout << "Call a function to draw a line" << endl; break;
        case RECT:
            cout << "Call a function to draw a rectangle" << endl; break;
        case CIRCLE:
            cout << "Call a function to draw a circle" << endl; break;
        default:
            cerr << "Error: Unsupported shape" << endl; break;
    }

    return 0;
}
```

Example: Mixing Colors

```
#include <iostream>     /* File: enum-colors.cpp */
using namespace std;

int main()
{   // Declare color variables immediately after the enum definition
    enum color { RED, GREEN, BLUE, YELLOW, CYAN, PURPLE } x, y;
    int xint, yint;     // Input variables for the color variables

    cin >> xint >> yint;
    x = static_cast<color>(xint); // Convert an int to a color quantity
    y = static_cast<color>(yint); // Convert an int to a color quantity

    if ( (x == RED && y == GREEN) || (y == RED && x == GREEN) )
        cout << YELLOW << endl;

    else if ( (x == RED && y == BLUE) || (y == RED && x == BLUE) )
        cout << PURPLE << endl;

    else if ( (x == GREEN && y == BLUE) || (y == GREEN && x == BLUE) )
        cout << CYAN << endl;

    else
        cerr << "Error: only support mixing RED/GREEN/BLUE!" << endl;

    return 0;
} // Check what is really printed out
```