Introduction to Object-Oriented Programming

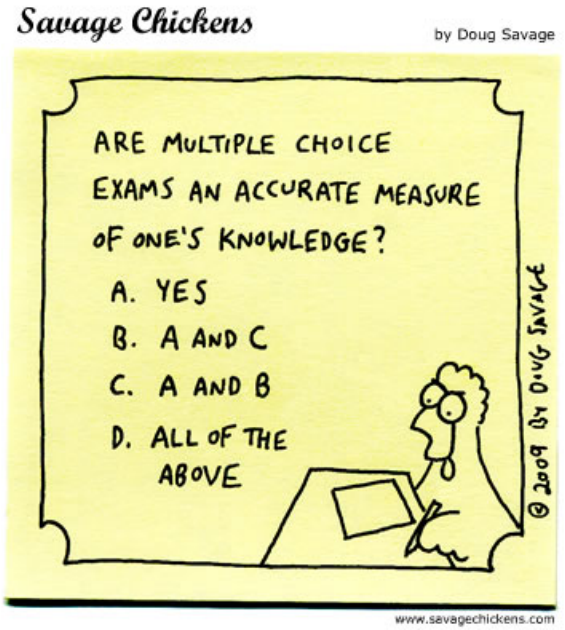
COMP2011: C++ Control II

Dr. Cindy Li
Dr. Brian Mak
Dr. Dit-Yan Yeung

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

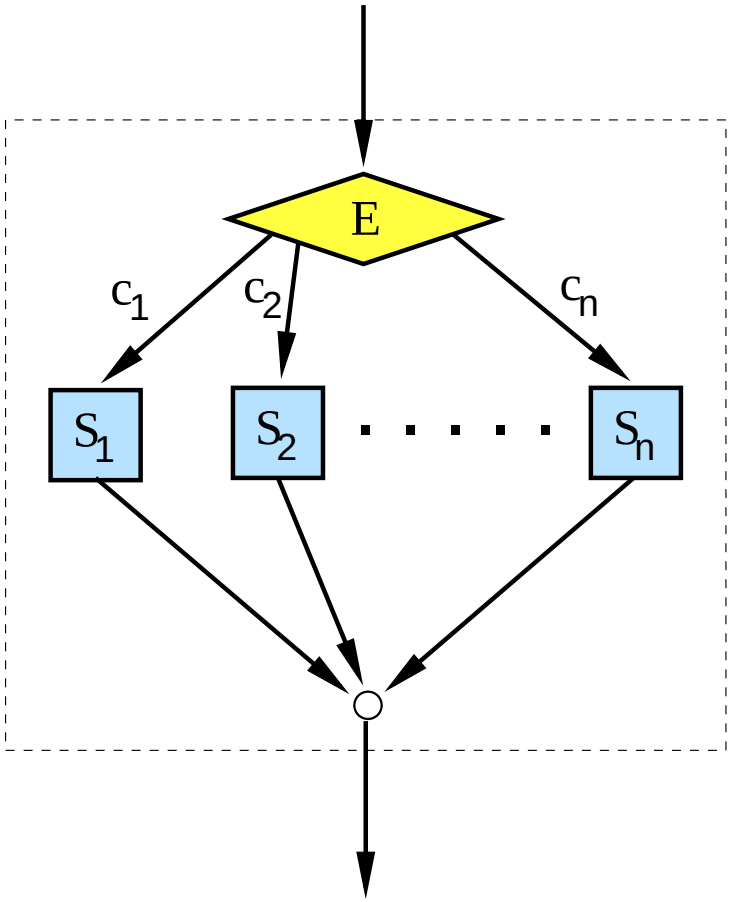
Part I

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(void)                                                    // To determine your instructor.
{
    int section;                                    // COMP2011 section number: should be 1,2, or 3

    cout << "Enter the COMP2011 section number to find its instructor:  ";
    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;
        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(void)                                               // To find out who may give you blood
{
    char bloodtype;
    cout << "Enter your blood type (put 'C' for blood type AB): ";
    cin >> bloodtype;

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

    return 0;
}
```

Example: switch with Sharing Cases

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

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

    switch (mark/10)
    {
        case 10:                                             // Several cases may share the same statements
        case 9:
            grade = 'A'; break;                              // If mark >= 90
        case 8:
        case 7:
        case 6:
            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(void)                                  /* To determine your grade. (fictitious) */
{
    int mark;                                   // Numerical mark between 0 and 100
    char grade;                                                    // Letter grade
    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(void)                                       // To determine your grade. (fictitious)
{
    int mark;                                        // Numerical mark between 0 and 100
    char grade;                                                          // Letter grade
    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: Mixing Colors

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

enum color { RED, GREEN, BLUE, YELLOW, CYAN, PURPLE, WRONG_COLOR = -1 };

color mix_colors(color x, color y)
{
    if ( (x == RED && y == GREEN) || (y == RED && x == GREEN) )
        return YELLOW;

    else if ( (x == RED && y == BLUE) || (y == RED && x == BLUE) )
        return PURPLE;

    else if ( (x == GREEN && y == BLUE) || (y == GREEN && x == BLUE) )
        return CYAN;

    else
    {
        cerr << "Error:  Currently we only support mixing RED/GREEN/BLUE!" << endl;
        return WRONG_COLOR;
    }
}

int main(void)
{
    cout << mix_colors(RED, GREEN) << endl;
    cout << mix_colors(RED, BLUE) << endl;
    cout << mix_colors(GREEN, BLUE) << endl;
    cout << mix_colors(GREEN, YELLOW) << endl;

    return 0;
}
```

Example: enum with switch

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

int main(void)
{
    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;
}
```