Module 15:
Switch Statements

Intro to Computer Science 1 - C++
Professor Scott Frees

Textbook

This topic is covered in section 3.13 of the text

Menu-based interface

Programs often use menus to present the user with several choices

To make a selection, the user can enter a single character

```
char choice;
cout << "Enter selection:  ";
cin   >> choice;
if (choice == 'y' ) {
  …
} ….
```

Programming Example 15

Ask the user if they want to compute the area of a Rectangle (r), Square (s), or Circle (c)

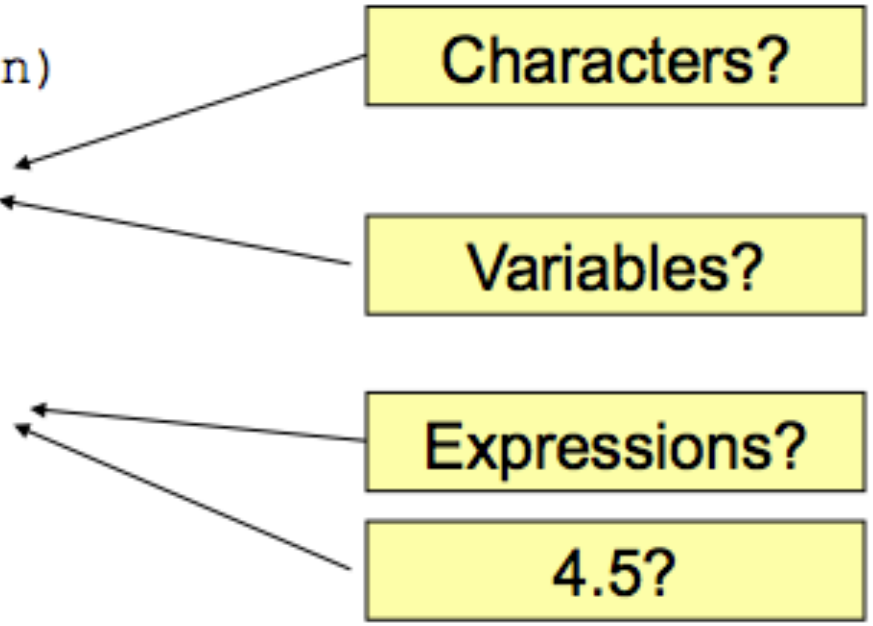
–For Rectangle, ask for width and height.

–For Square, ask for width only

–For Circle ask for radius

Switch Statement

```
switch (IntegerExpression)
{
   case IntegerConstant:
   // statements
       break;
   case IntegerConstant:
   // statements
       break;
   …
   default:
       // execute if no cases are a match
}
```



Switch Statement Criteria

1) You are always comparing the **same** expression against a series of choices
2) You are always comparing with equality (==)
3) Your choices are integer based (int, short, char…)
4) Your choices are mutually exclusive

Switch and If

```
if ( x == 5) {
    cout << "1" << endl;
}
else if (x == 6){
    cout << "2" << endl;
}
else if (x == 7) {
    cout << "3" << endl;
}
else {
    cout << "The rest" << endl;
}
```

```
switch ( x ) {
case 5:
    cout << "1" << endl;
    break;
case 6:
    cout << "2" << endl;
    break;
case 7:
    cout << "3" << endl;
    break;
default:
    cout << "The rest" << endl;
}
```

Menu-based interface - redux

Using switch statements for determining the user's choice is very convenient.

Program: Rewrite Area calculator using switch statement

Pitfalls when using switch

- Be ***extremely*** careful to not omit `break` when it should be there!
  - There are times omitting it is actually quite useful however.
  - If **omitted**, always add comment explaining why.

- You cannot compare two *variables* with a switch - you can only compare against **integer or character literals**
  - Booleans also work, but not that useful