Lecture 4

Chapter 2.2,2.3
Topics for today

• if – else statements
• Multiple statement if – else
• Multiway if – else
• Switch statements
• Enumeration types
• Conditional operator
• While
• do while
• Comma expression
• Empty statement
• Nested loops
• Infinite loop
• Break and continue
if - else

if(grade > 90)
    cout << “You got an A!!”;
else
    cout << “You didn’t get an A!!”;}
if

if(grade > 90)
    cout << "You got an A!!";
if – else multiple statements

```cpp
if(grade > 90)
{
    cout << “You got an A!!”;
    GPA = 4.0;
}
else
{
    cout << “You didn’t get an A!!”;
}
```
if – else multiway

if ( grade > 90)
    cout << “You got an A!!”;
else if ( grade > 80)
    cout << “You got an B!!”;
else if ( grade > 70)
    cout << “You got an C!!”;
switch statements

switch (grade)
{
    case 97:
        cout << “You got an A+!!”;
        break;
    case 93:
        cout << “You got an A!!”;
        break;
    default:
        cout << “You got an A-!!”;
switch statements (corrected)

switch (grade) {
    case 100:
    case 99:
    case 98:
    case 97:
        cout << "You got an A+!!";  
        break;
    case 96:
    case 95:
    case 94:
    case 93:
        cout << "You got an A!!";  
        break;
    default:
        cout << "You got an A-!!";
}
Enumeration types

• Is a type whose values are defined by a list of constants of type int.
• These are for readability.
• For example
  – Instead of labeling directions 0,1,2,3
    • And telling a function that you are moving in direction 0
  – enum Direction{ NORTH, SOUTH, EAST, WEST};
Enumeration types

• Enumerations can be defined
  
  – enum MonthLength { JAN_LENGTH = 31, FEB_LENGTH = 28, MAR_LENGTH = 31, ... };
Conditional Operators

• Instead of
  – if (n1 > n2)
    max = n1;
  else
    max = n2;

• Conditional Operators
  – Max = (n1 > n2) ? n1 : n2;

• But the book does not advise using them
While loops

- `while(++x < 10)`
  ```
  {
    cout << x << endl;
  }
  ```
- What if you want to print the first value of x?
While loops

• do
  
  {  
      cout << x << endl;
  }
  
while(++x < 10)

• This will run the code once before checking the condition
Comma expression

• What is the value of
  – result = (first = 2, second = first + 1);

• These are useful for for loops but bad programming practice otherwise.
for loops

- for(initialization; boolean_expression; update)

```java
sum = 0;
for(int n = 1; n <= 10; n++)
    sum += n;
```

```java
for( sum = 0, int n = 1; n <= 10; n++)
    sum += n;
```
Nested for loops

sum = 0;
for(int n = 1; n <= 10; n++)
    for(int m = 20; n <= 100; m++)
        sum += n;
Empty statement

• for(int count = 1; count <= 10; count++);
  – This for loop won’t do anything but it is legal code

• Previous example could be written as
  – for(sum = 0, n = 1; n <= 10; sum += n, n++);
Pitfalls with loops

for( float x = 0.; x != 12.; x += (1./3.))

for( int x = 1; x != 6; x += 2)
Continue and Break

```cpp
while( miles++ < range )
{
    if( areWeThereYet )
        break;
    else
        continue;
}
```
Continue and Break

• Continue – ends the current loop iteration of the nearest loop
• Break – breaks the flow out of the nearest loop