CS 102 Introduction to Programming Using C++

Chapter 4

Looping

Homework and Programs

- Homework
- R4.1, 2, 3, 5, 7, 8, 10, 12, 13, 23, 24

- Programs
- p. 178, P4.1, 2, or 3.
- Also one of P4.18, 19, 20, and 25.

Academic Honesty

- As a reminder, make sure <u>you</u> write all of your own code
- This is our academic honesty policy:
  - All programs must be written by you alone
- Exceptions
  - You may base your programs on programs in the textbook
  - You can get help from me

Homework and Programs

- Homework
- R4.1, 2, 3, 5, 7, 8, 10, 12, 13, 23, 24
- Programs
- p. 178, P4.1, 2, or 3.
- Also one of P4.18, 19, 20, and 25.

Structured Programming (Again)

- The three structures of structured programming are
  1. Sequence
  2. Decision
  3. Repetition

Repeating Commands

- Sometimes you want to do something repeatedly
  - Doing something repeatedly is called looping
- C++ has several ways to do that
- One way is the <u>while</u> statement

Counting to Ten

- Here is a <u>while</u> loop that counts to 10

```
int i = 1;
while (i <= 10)
{
    cout << i << endl;
    i++;
}
```

An Example from the Textbook

```
const double RATE = 5;
const double INITIAL_BALANCE = 10000;
const double TARGET = 2 * INITIAL_BALANCE;

double balance = INITIAL_BALANCE;
int year = 0;

while (balance < TARGET)
{
  year++;
  double interest = balance * RATE / 100;
  balance = balance + interest;
}
```

Variables

This statement appears inside the loop

```
double interest = balance * RATE / 100;
```

It creates the double variable interest

Since the variable is declared inside the loop, it is actually re-created each time through the loop

One time through the loop is called an iteration

- So we can say the variable is re-created on each iteration

In addition, the variable is destroyed after the loop

This means it’s as if the variable was never declared

Hand Tracing

- Let’s trace the code to see how it works

A Faulty Loop

- Let’s look again at the while loop that counts to 10

```
int i = 1;
while (i <= 10)
{
     cout << i << endl;
}
```

- Here I have deleted one line
- This is an example of an infinite loop

Loop Exercise from the Textbook

- Can you find the logic error in this program fragment?

```
int n = 1;
while (n != 50)
{
    cout << n << endl;
    n = n + 10;
}
```

Hand Tracing a While Loop

- How does this code count?
- Assume upper_limit is a given integer variable.

```
int i = 1;
while (i < upper_limit)
{
    cout << i << endl;
    i++;
}
```

Hand Tracing Another While Loop

- How does this code count?
- Assume upper_limit is a given integer variable.

```
int i = 1;
while (i < upper_limit)
{
    i++;
    cout << i << endl;
}
```

A Pre-Test Loop

- The while loop is an example of a pre-test loop
    - The condition is checked before the code in the loop is run
- This means that the loop can be skipped entirely!
- Check this example

An Example of a Pre-Test Loop

```
int i = 50;
while (i < 10)
{
    cout << i << endl;
    i++;
}
```

A Second Type of Loop

- Another type of loop is the <u>for</u> loop
- The <u>for</u> loop counts
    - It is sometimes called a counter-controlled loop

An Example of a for Loop

```
int i;
for (i=1;  i<=10;  i++)
{
    cout << i << endl;
}
```

Declaring the Counter Variable at the Last Minute

- You can declare the counter variable right inside the <u>for</u> loop
- This version is very common

```
for (int i=1;  i<=10;  i++)
{
    cout << i << endl;
}
```

- Why would people code it this way?

Counting up, Counting down, …

- The <u>for</u> loop can count in many ways
- Some examples follow

Counting up

```
for (int i=2;  i<11;  i=i+2)
{
    cout << i << endl;
}

for (int i=10;  i<51;  i=i+10)
{
    cout << i << endl;
}
```

Counting down

```
for (int i=10;  i>0;  i– –)
{
    cout << i << endl;
}

for (int i=51;  i>0;  i=i-10)
{
    cout << i << endl;
}
```

C++ Has a Character Type

- Let's take a sneak peek at Chapter 7
- C++ has a type that can hold a single character
  - A string can certainly do this, but it stores data in a different way
  - The string is more wasteful for storing a single character

Focusing on that Character Type

- The type is char
- You use single quotes (like ') for char data

  char middle_init = ‘A’

- Don’t forget
  - You use double quotes (like ") for string data

The string Type vs. the char Type

```
string name;
name = "Harry";
```

- The string requires extra memory to keep track of where it ends

```
char middle_initial;
middle_initial = 'T';
```

The string Type vs. the char Type Part 2

- A string is just a list of characters
- We can code like this

```
string song = "By the Beautiful Blue Danube";
int num_caps = 0;
for (int i=0;  i<song.length();  i++)
{
     if  (song [i] >= 'A'  &&  song [i] <= 'Z')
          num_caps++;
}
```

Pre-Test Loops Again

- The <u>for</u> loop is another example of a pre-test loop
- The condition is checked before the loop is run
- As with the <u>while</u> loop, this loop can be skipped (not executed at all)
- This is useful for counting items

Counting Items

- You read in several items with a while loop
- You process them with a <u>for</u> loop
- What if there are no items to process?

```
int last = –1;
for (int i=0;  i<last;  i++)
{
        … // Lots of lines deleted here
}
```

A Third Type of Loop

- This is the <u>do</u> loop
- An example is

```
int i=1;
do
{
    cout << i << endl;
    i++;
}
while (i<10);
```

A Post-Test Loop

- This might be obvious from the layout of the code
- The do loop is a post-test loop
- The condition is tested after the body of the loop is run
  - The body of a loop (any loop) is the code that is run on each iteration of the loop
- This means the do loop always iterates at least once

An Example of a Post-Test Loop

```
int i = –1;
do
{
    cout << i << endl;
    i++;
}
while (i>0);
```

Input Validation

- A good use of a post-test loop is input validation

```
int value;
do
{
    cout << "Enter a non-negative integer: ";
    cin >> value;
}
while (value < 0);
```

Validating Input with a while Loop

- Let’s try this with a while loop
- Why won’t similar code work?
- How would we have to change it?

```
int value;
while (value < 0)
{
    cout << "Enter a non-negative integer: ";
    cin >> value;
}
```

Questions?

- Are there any questions?