EECE.2160: ECE Application Programming

Summer 2018

Lecture 4: Key Questions
May 30, 2017

QUESTIONS:

1. Explain the usage and basic structure of a `while` loop.

2. Explain how `while` loops can be used:
a. When number of iterations is dependent on a variable (flexible limit) (`while2.c`)

b. When you want to repeat an operation until a given value (sentinel) is entered (`while3.c`)

3. What is the difference between a `while` loop and a `do-while` loop?

4. In what cases are `for` loops useful? Describe the basic structure of a `for` loop.

5. Describe the operators that allow you to directly modify a variable without writing a full assignment statement.

6. Explain the difference between pre- and post-increment or decrement operators.

EXAMPLES:

1. What does each of the following short programs print?

a.
```
x = 7;
while ( x < 10  )
{
   printf("%d ",x);
   x = x + 1;
}
```

b.
```
x = 7;
while ( x < 3  )
{
   printf("%d ",x);
   x = x + 1;
}
```

2. Finish the following program as directed

```
int main() {
    int i;                    // Number to square
    int iSquared;             // Square of the number
    printf(" i       i^2\n"); // Column headings

    // Compute and display the squares of numbers 0 to 10
    // Use a field width of 2 to print i and 10 to print i^2
    //   with no extra space between the fields









    return 0;
}
```

3. Show the difference between the outputs of the loops below

```
x = 7;
do {
     printf("%d",x);
     x = x + 1;
} while ( x < 3  );
```

```
x = 7;
while ( x < 3  )
{
     printf("%d",x);
     x = x + 1;
}
```

4. Recall the example for using a while loop with a sentinel value in the grade average program and show that loop written as a `do-while` loop.

5. What does the following program print?

```
int n = 5;
printf("n = %d\n", ++n);
printf("Now, n = %d\n", n++);
printf("Finally, n = %d\n", n);
```

6. **Example:** What does each of the following print?

a.
```
for (i = 5; i < 40; i += 8)
{
    printf("%d ", i);
}
```

b.
```
for (i = -5; i < -10; i--)
{
  printf("%d ", i);
}
```

c.
```
for (i = 10; i <= 100; i = i+10)
{
    if (i % 20)
        printf("%d ", i);
}
```

d.
```
for (i = 5; i < 10; i += i%2)
{
  printf("%d ", i++);
}
```