16.216: ECE Application Programming

Summer 2012

Lecture 5: Key Questions

July 26, 2012

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

2. **Example:** 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;
   }
```

3. **Example:** 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;
}
```

4. 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`)

```c
/*
 * while2.c
 * Adapted from earlier solution by Prof. George Cheney
 * 16.216: ECE Application Programming
 * ECE Dept., UMass Lowell
 *
 * PURPOSE: Read list of grades from keyboard and compute average
 *
 * DEMONSTRATES: A counting loop with a flexible limit
 */

#include <stdio.h>

int main()
{
    int numGrades;      // Requested # of grades
    int gradeCount;     // Counts # of grades processed so far

    double grade;       // An individual grade to be processed
    double gradeSum;    // Running total
    double avgGrade;    // Average grade

    // Prompt for and read # of grades
    printf("How many grades? ");
    scanf("%d", &numGrades);

    // Prompt user to enter grades
    printf("Enter %d grades:\n", numGrades);

    // Initialize loop
    gradeSum = 0;
    gradeCount = 0;

    // Repeatedly read grades until limit is reached
    while (gradeCount < numGrades) {
        scanf("%lf", &grade);           // Read grade and accumulate it
        gradeSum = gradeSum + grade;

        gradeCount = gradeCount + 1;    // Increment grade count
    }

    // Compute and display the average
    avgGrade = gradeSum / numGrades;
    printf("Average grade = %1.1lf\n", avgGrade);
    return 0;
}
```

```c
/*
 * while3.c
 * Adapted from earlier solution by Prof. George Cheney
 * 16.216: ECE Application Programming
 * ECE Dept., UMass Lowell
 *
 * PURPOSE: Read list of grades from keyboard and compute average.
 *   Program will run until user enters invalid grade (something
 *   outside the range 0 <= grade <= 100), which is taken as
 *   signal that
 *
 * DEMONSTRATES: Loop terminated on sentinel value
 */

#include <stdio.h>

int main()
{
    int gradeCount;     // Counts # of grades processed so far

    double grade;       // An individual grade to be processed
    double gradeSum;    // Running total
    double avgGrade;    // Average grade

    char enterGrade;

    // Initialize loop
    gradeSum = 0;
    gradeCount = 0;

    // Prompt for and read first grade
    /*printf("Enter grade: ");
    scanf("%lf", &grade);*/

    // Continue reading/accumulating grades until invalid value entered
    do {
        printf("Enter grade: ");        // Prompt for and read next grade
        scanf("%lf", &grade);

        // If invalid grade entered, leave loop
        if ((grade > 100) || (grade < 0))
            break;

        gradeSum = gradeSum + grade;    // Accumulate grade
        gradeCount = gradeCount + 1;    // Increment grade count

    } while ((grade >= 0.0) && (grade <= 100.0));

    if (gradeCount == 0)                // No grades entered
        printf("No valid grades entered\n");

    // Compute and display the average
    else {
        avgGrade = gradeSum / gradeCount;
        printf("Average grade = %1.1lf\n", avgGrade);
    }

    return 0;
}
```

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

6. 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;
}
```

7. 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.

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

a.
```
int i = 0;
while (i < 30) {
   if ((i % 3) == 0)
         printf(“%d\n”, i);
   i = i + 2;
}
```

b. *Assume input is: 3 8 -4 -7 0 5 2*
```
do {
   scanf(“%d\n”,&x);
   printf(“x = %d, x/2 = %d\n”, x, x/2);
} while ((x > 2) || (x < 0));
```

c. *Assume input is: b c d e f g h*
```
char c = ‘a’;
while (c != ‘h’) {
   printf(“%c”, c);
   scanf(“%c”, &c);
}
```

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

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

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

12. **Example:** 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);
```

13. **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++);
}
```

14. Explain the use of `break` and `continue` statements with loops.