16.216: ECE Application Programming

Summer 2015

Lecture 3: Key Questions
May 26, 2015

1. Explain the basic form of an `if` statement.

2. Describe how the expression in `if (<expression>)` is evaluated and show how conditions are evaluated, including multiple conditions in the same expression.

3. Describe how the statement—the actual code to be executed if the condition is true—is written for an `if` statement.

4. Show how multiple if statements can be nested together (`if/else if/else`).

5. **Example:** What does the following code print?

```
int main() {
    int x = 3;
    int y = 7;

    if (x > 2)
        x = x - 2;
    else
        x = x + 2;

    if ((y % 2) == 1)
    {
        y = -x;
        if ((x != 0) && (y != -1))
           y = 0;
    }
    printf("x = %d, y = %d\n", x, y);
    return 0;
}
```

6. **Example:** Write a short code sequence that does each of the following:

a. Given `int x`, check its value. If `x` is more than 5 and less than or equal to 10, print `x`

b. Prompt for and read temperature as input (type `double`). If temp is 90 or higher, print "It's too hot!" If temp is 32 or lower, print "It's freezing!" In all other cases, print "It's okay"

c. Read 3 `int` values and print error if input problem

- If fewer than 3 values read, print error message with number of values
- Example: `Error: only 2 inputs read correctly`

7. Describe the basic format of a `switch` statement, including its general usage, the use of `case` and `default`, and the use of the `break` statement.

8. Describe a situation in which you might not want to use a `break` statement at the end of a given `case`.

9. **Example:** Given the code below:

```
int main() {
      char grd;

      printf("Enter Letter Grade: ");
      scanf("%c",&grd);
      printf("You are ");

      switch (grd)      {
      case 'A' :
            printf("excellent\n");
            break;
      case 'B' :
            printf("good\n");
            break;
      case 'C' :
            printf("average\n");
            break;
      case 'D' :
            printf("poor\n");
            break;
      case 'F' :
            printf("failing\n");
            break;
      default :
            printf("incapable of reading directions\n");
            break;
      }
      return 0;
}
```

What does the program print if the user inputs:

a. A

b. B+

c. c

d. X

10. How could we easily change each case to recognize both upper and lowercase inputs?

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

12. **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;
   }
```

13. **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;
}
```

14. 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`)

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

16. 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;
}
```

17. 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.