EECE.2160: ECE Application Programming

Spring 2016

Lecture 9: Key Questions
February 8, 2016

1. 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.

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

3. **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

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

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

6. **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;
}
```

7. **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;
}
```

8. 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`)

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

10. 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;
}
```

11. 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.