

## Unit Four Lecture One:

### Topic 1: Iteration and the while loop

Def. of Iteration: The ability to repeat sections of code an arbitrary number of times.

Def. of while loop: The statements within the body of the loop are executed over and over until the condition, in the parentheses following the key word *while*, becomes false. The condition is in the loop header statement.

It takes the general form...

```
while ( condition(s) )
{
    statement sequence; // body of the loop
}
```

### Topic 2: Counters

Def. of counter: A variable that is incremented by one every time it is encountered.

examples: 1) `c = c + 1;`  
          2) `c += 1;`  
          3) `c++;`           // this is most popular

[`c--;` decrements `c` by one / same as `c = c - 1`]

### Topic 3: Accumulators

Def. of accumulator: A variable that sums numbers as they are entered.

examples: 1) `a = a + n;`  
          2) `a += n;`       // this is most popular

[`a -= n;` subtracts `n` from `a` / same as `a = a - n`]

#### Topic 4: Specific examples of while loops

1) The count controlled loop

```
int count = 1;
int accum = 0;

while (count <= 5)
{
    accum += count;
    count++;
}
```

What are the values of the variables when we drop out of the loop?

count = \_\_\_\_\_ accum = \_\_\_\_\_

2) The sentinel controlled loop

```
int num;
int accum = 0;
int count = 0;
String input;

input = JOptionPane.showInputDialog("Enter Number: ");
num = Integer.parseInt(input);

while (num != -1)
{
    count++;
    accum += num;
    input = JOptionPane.showInputDialog("Enter Number: ");
    num = Integer.parseInt(input);
}

System.out.println("Sum = " + accum);
```

If the inputted data is as follows: 5, 8, 1, 6, -1

What are the values of the variables when we drop out of the loop?

count = \_\_\_\_\_ accum = \_\_\_\_\_

3) The simple event controlled loop

```
int a = 10;
int b = 6;
int dif, temp;

dif = a - b;

while (dif <= 10)
{
    temp = b;
    b = a;
    a = 2 * temp;

    dif = a - b;
}

System.out.println("a = " + a + " b = " + b);
```

What are the values of the variables when we drop out of the loop?

a = \_\_\_\_\_ b = \_\_\_\_\_ dif = \_\_\_\_\_ temp = \_\_\_\_\_

4) The flag controlled loop

```
int i = 1;
int accum = 0;
int val = 3;
boolean flag = true;

while ((i<=5) && (flag))
{
    val += 2;
    if (val > 5) accum += val;
    if (accum > 12) flag = false;

    i++;
}
```

What are the values of the variables when we drop out of the loop?

i = \_\_\_\_\_ accum = \_\_\_\_\_ val = \_\_\_\_\_ flag = \_\_\_\_\_

## Topic 5: \t - The Tab command

The \t uses fields of width 8 characters.

Example:

```
JTextArea output = new JTextArea();
output.setFont(new Font("Monospaced", Font.BOLD, 12));
output.setText("12345678901234567890123456789\n");
output.append("a =\t" + 5 + "\tb =\t" + 12);
```

Output for this code segment would look like this...

```
12345678901234567890123456789
a =      5      b =     12
```

If a tab field of 8 characters is not appropriate for your specific application, the size of the tab field can be changed.

Example:

```
JTextArea output = new JTextArea();
output.setFont(new Font("Monospaced", Font.BOLD, 12));
output.setText("12345678901234567890123456789\n");
output.setTabSize(6);
output.append("a =\t" + 5 + "\tb =\t" + 12);
```

Output for this code segment would look like this...

```
12345678901234567890123456789
a =   5      b =   12
```

## Topic 6: The Math class

The Math class which is included in the standard Java library is frequently useful. The Math class provides a range of common mathematical methods, some of which are listed below. Inputs and outputs for most of these methods are doubles.

```
1) double a = Math.sqrt(num);           // returns square root of num
2) double b = Math.pow(x, y);           // returns xy
3) long    c = Math.round(num);         // returns num rounded to the
                                         // nearest whole number
4) double d = Math.abs(num);            // returns absolute value of num
5) double e = Math.ceil(x);             // returns smallest integer ≥ x
```

You will notice that the call of each of these methods is *class.method()* instead of *object.method()*, this means that all of these methods are *static*. We will cover this topic in more detail later in the course.

Assignment U4A1: Converting Binary Numbers