Brendan Whitaker

CSE 2221 Homework 3

Professor Bucci

1/18/2017

1.

a.

```
int i = 0;
while ((i*i) < n) {
        out.print(i*i + " ");
        i++;
}
```

b.

```
int i = 1;
while ((10*i) < n) {
        out.print(10*i + " ");
        i++;
}
```

c.

```
//we assume non-negative integer powers of n.
int i = 0;
while ((Math.pow(2, i)) < n) {
        out.print(Math.pow(2, i) + " ");
        i++;
}
```

3. We rewrite the given for loop into a while loop:

```
int i = 1, s = 0;
while (i <= 10) {
        s += i;
        i++;
}
```

4. The following snippet of code sums the first n terms in the Gregory-Leibniz series:

```
int i = 0;
while (i <= n) {
        pi += Math.pow(-1, i)/((2*n) + 1);
        i++;
}
pi *= 4;
```

5. The following snippet of code assigns to sum the result of adding up all integers of the form $n^2 + m^2$ where:

- both n and m are at least 1
- $n^2 <$ areaBound
- $m^2 <$ areaBound

```
int n = 1;
int m;
while (n*n < areaBound) {
        m = 1;
        while (m*m < areaBound) {
                sum += (n*n) + (m*m);
                m++;
        }
        n++;
}
```

6. The following snippet of code keeps adding terms until the difference between two consecutive estimates is less than some predefined tolerance, say double epsilon = 0.0001:

```
int i = 0;
double currentTerm = 1.0;
while (currentTerm > epsilon) {
        currentTerm = Math.pow(-1, i)/((2*n) + 1);
        pi += currentTerm;
}
pi *= 4;
```