

*C++ Primer Plus, 5<sup>th</sup> Edition* by Stephen Prata  
Chapter 5: Loops and Relational Expressions  
Review Questions

1. What's the difference between an entry-condition loop and an exit-condition loop? Which kind is each of the C++ loops?

*An entry-condition loop evaluates the test-condition before the first execution of the body. An exit-condition loop evaluates the test-condition after the first execution of the body. Thus, an entry-condition loop executing at least once is contingent on the test-condition evaluating to **true** the first time and an exit-condition loop will always execute the body at least once, even if the test-condition is false. **for** and **while** loops are entry-condition loops and **do while** loops are exit-condition loops.*

2. What would the following code fragment print if it were part of a valid program?

```
int i;
for (i = 0; i < 5; i++)
    cout << i;
    cout << endl;
```

*The code would print the following:*

01234

3. What would the following code fragment print if it were part of a valid program?

```
int j;
for(j = 0; j < 11; j +=3)
    cout << j;
cout << endl << j << endl;
```

*The code would print the following:*

0369

12

4. What would the following code fragment print if it were part of a valid program?

```
int j = 5;
while (++j < 9)
    cout << j++ << endl;
```

*The code would print the following:*

6

8

5. What would the following code fragment print if it were part of a valid program?

```
int k = 8;
do
    cout << "k = " << k << endl;
while (k++ < 5);
```

*The code would print the following:*

```
k = 8
```

6. Write a `for` loop that prints the values 1 2 4 8 16 32 64 by increasing the value of a counting variable by a factor of two in each cycle.

*See the following code:*

```
for (int i = 1; i < 65; i *= 2)
    cout << i << " ";
```

7. How do you make a loop body include more than one statement?

*To make a loop body include more than one statement, braces should immediately follow the loop and enclose whichever statements are to be included in the body of the loop.*

8. Is the following statement valid? If not, why not? If so, what does it do?

```
int x = (1,024);
```

What about the following?

```
int y;
y = 1,024;
```

*Both statements are valid. The first statement assigns the value 024 to x. Since 024 is octal, it is converted to an int which would be 20 in decimal. The second statement initializes y to 1. The comma operator separates two or more expressions where only one is expected. The comma operator has the lowest precedence among any C++ operators. When the set of expressions has to be evaluated for a value, only the rightmost expression is considered.*

9. How does `cin>>ch` differ from `cin.get(ch)` and `ch=cin.get()` and how it views input?

*cin>>ch skips over whitespace, where the other two do not.*