**WEEK 8 – Review Questions and**

**Programming Challenges Handout**

**Chapter: 5**

**Review Questions:**

**Short Answers:**

1. Write code that lets the user enter a number. The number should be multiplied by 2 and printed until the number exceeds 50. **Use a while loop.**

int number;

cout << “Enter a number: “;

cin >> number;

while(number <= 50){

number = number \* 2;

}

cout << number;

1. **Convert the following while loop to a for loop:**

int count = 0;

while (count < 50)

{

cout << "count is " << count << endl;

count++;

}

for(int i = 0; i < 50; i++){

cout << “count is “ << i << endl;

}

1. **Convert the following for loop to a while loop:**

for (int x = 50; x > 0; x--)

{

cout << x << " seconds to go.\n";

}

int counter = 50;

while(counter > 0){

cout << counter << “ seconds to go.\n”;

counter--;

}

**Programming Challenges:**

1. **Hotel Suites Occupancy**

Write a program that calculates the occupancy rate of the 120 suites (20 per floor) located on the top 6 floors of a 15-story luxury hotel. These are floors 10–12 and 14–16 because, like many hotels, there is no 13th floor. Solve the problem by using a *single* loop that loops once for each floor between 10 and 16 and, on each iteration, asks the user to input the number of suites occupied on that floor. Use a nested loop, loop to validate that the value entered is between 0 and 20. After all the iterations, the program should display how many suites the hotel has, how many of them are occupied, and what percentage of them are occupied.

1. **Sales Bar Chart**

Write a program that asks the user to enter today’s sales rounded to the nearest $100 for each of three stores. The program should then display a bar graph comparing each store’s sales. Create each bar in the graph by displaying a row of asterisks. Each asterisk should represent $100 of sales.

Here is an example of the program’s output. User input is shown in bold.

Enter today’s sales for store 1: **1000[Enter]**

Enter today’s sales for store 2: **1200[Enter]**

Enter today’s sales for store 3: **900[Enter]**

**DAILY SALES**

**(each \* = $100)**

Store 1: \*\*\*\*\*\*\*\*\*\*

Store 2: \*\*\*\*\*\*\*\*\*\*\*\*

Store 3: \*\*\*\*\*\*\*\*\*

1. **Savings Account Balance**

Write a program that calculates the balance of a savings account at the end of a three month period. It should ask the user for the starting balance and the annual interest rate. A loop should then iterate once for every month in the period, performing the following steps:

**A)** Ask the user for the total amount deposited into the account during that month and add it to the balance. Do not accept negative numbers.

**B)** Ask the user for the total amount withdrawn from the account during that

month and subtract it from the balance. Do not accept negative numbers or

numbers greater than the balance after the deposits for the month have been

added in.

**C)** Calculate the interest for that month. The monthly interest rate is the annual

interest rate divided by 12. Multiply the monthly interest rate by the average of

that month’s starting and ending balance to get the interest amount for the

month. This amount should be added to the balance.

After the last iteration, the program should **display a report that includes the following information:**

**•** starting balance at the beginning of the three-month period

**•** total deposits made during the three months

**•** total withdrawals made during the three months

**•** total interest posted to the account during the three months

**•** final balance