Lab 16 (from professor Kent Chin)

1) Provide the output of the given line in the blanks below.

#include <iostream>

using namespace std;

int fun1(int n) {
    return n*n;
}

int fun2(int x, int y) {
    if (x > y) return x;
    return y;
}

int fun3(int a, int b, int c) {
    return (a*b) + c;
}

int main() {
    int x = 4, y = 7;
    cout << fun1(x) << endl; //(1)___________________________
    cout << fun1(x+y) << endl; //(2)___________________________
    cout << fun1(y+6) << endl; //(3)___________________________
    cout << fun2(x,y) << endl; //(4)___________________________
    cout << fun2(5,-4) << endl; //(5)___________________________
    cout << fun2(y, 8) << endl; //(6)___________________________
    cout << fun3(1,5,2) << endl; //(7)___________________________
    cout << fun3(x,y,x) << endl; //(8)___________________________
    cout << fun3(y,4,-4) << endl; //(9)___________________________

    return 0;
}

2) Write a function called cube which takes an integer parameter by reference, and cubes the parameter:

Sample run
int x = 2;
cout << x << endl; //2
cube(x);
cout << x << endl; //8
3) (from p.114 in Schaum's Programming with C++) Write and test the following computeCircle() function that returns the area a and the circumference c of a circle with given radius r:

```
void computeCircle(float& a, float& c, float r)
```

Note: You can set a variable name PI to 3.141592653589793.

4) Write a complete C++ program that does the following: (no need to use reference parameters)

1. The program asks user to enter a positive integer n < 10. If the input value is incorrect, the program repeatedly force user to input value until n is within the expected value.

2. The program then invoke (calls) function fun1( ) that does the following:

   i. The heading of the function is void fun1(int n)

   ii. The function prints a square with n rows and n columns using the letter X on or above the main diagonal and the letter O below it.

For example, if the user enters 4 for n the output is as follows:

```
XXXX
OXXX
OOXX
OOOX
```