University of Management and Technology

UNIVERSITY OF MANAGEMENT AND TECHNOLOGY LAHORE

Assignment # 5 (Theory)

Course Title: Programming Fundamentals
Program Name: BS(CS), BS(SE)
Course Instructor: Amjad Ali
Semester: Fall 2015
Marks: 100 points
Due Date: 14-01-2016

Question No. 1

Consider the following statements:

```
double num1, num2, num3;
int int1, int2, int3;
int value;

num1 = 5.0; num2 = 6.0; num3 = 3.0;
int1 = 4; int2 = 7; int3 = 8;
```

and the function prototype:

double cube(double a, double b, double c);

Which of the following statements are valid? If they are invalid, explain why.

a. value = cube (num1, 15.0, num3); **Valid**

b. cout << cube(num1, num3, num2) << endl; **Valid**

c. cout << cube(6.0, 8.0, 10.5) << endl; **Valid**

d. cout << cube(num1, num3) << endl; **Invalid (signature mismatched)**

e. value = cube(num1, int2, num3); **Invalid (signature mismatched)**

f. value = cube(7, 8, 9); **Invalid (signature mismatched)**

g. value = cube(int1, int2, int3); **Invalid (signature mismatched)**

Question No. 2

Consider the following functions:

```
int secret(int x)
{
        int i, j;
        i = 2 * x;
        if (i > 10)
                j = x / 2;
        else
                j = x / 3;
return j - 1;
}

int another(int a, int b)
{
        int i, j;
        j = 0;
        for (i = a; i <= b; i++)
                j = j + i;
return j;
}
```

What is the output of each of the following program segments? Assume that x, y, and k are int variables.

a. `x = 10;`
   `cout << secret(x) << endl;` **4**

b. `x = 5; y = 8;`
   `cout << another(x, y) << endl;` **26**

c. `x = 10; k = secret(x);`
   `cout << x << " " << k << " " << another(x, k) << endl;` **10 4 0**

d. `x = 5; y = 8;`
   `cout << another(y, x) << endl;` **0**

Question No. 3

Consider the following function prototypes:

```
int test(int, char, double, int);
double two(double, double);
char three(int, int, char, double);
```

Answer the following questions.

a. How many parameters does the function test have? What is the type of the function test ?
N0. Of parameters = 4 **Function Type = int**

b. How many parameters does function two have? What is the type of function two?
No. Of parameters = 2 **Function Type = double**

c. How many parameters does function three have? What is the type of function three?
No. Of parameters = 4 **Function Type = char**

d. How many actual parameters are needed to call the function test? What is the type of each actual parameter, and in what order should you use these parameters in a call to the function test?

No. Of actual parameters needed = 4

Parameter 1 = int type
Parameter 2 = char type
Parameter 3 = double type
Parameter 4 = int type

Same order as in the prototype

e. Write a C++ statement that prints the value returned by the function test with the actual parameters 5, 5, 7.3, and 'z'.

```
cout<<test(5, 'z', 7.3, 5);
```

f. Write a C++ statement that prints the value returned by function two with the actual parameters 17.5 and 18.3, respectively.

```
cout<<two(17.5, 18.3);
```

g. Write a C++ statement that prints the next character returned by function three. (Use your own actual parameters.)

```
cout<< static_cast<char> (three(5, 4, 'D', 7.3)+1);
```

Question No. 4

Consider the following function:

```
int mystery(int x, double y, char ch)
{
        int u;
        if ('A' <= ch && ch <= 'R')
                return(2 * x + static_cast<int>(y));
        else
                return(static_cast<int>(2 * y) - x);
}
```

What is the output of the following C++ statements?

a. cout << mystery(5, 4.3, 'B') << endl; **14**

b. cout << mystery(4, 9.7, 'v') << endl; **15**

c. cout << 2 * mystery(6, 3.9, 'D') << endl; **30**

Question No. 5

Consider the following function:

```
int secret(int one)
{
        int i;  int prod = 1;
        for (i = 1; i <= 3; i++)
                prod = prod * one;
return prod;
}
```

a. What is the output of the following C++ statements?

i. cout << secret(5) << endl; **125**

ii. cout << 2 * secret(6) << endl; **432**

b. What does the function secret do?

Question No. 6

Mark the following statements as true or false.

a. A function that changes the value of a reference parameter also changes the value of the actual parameter. **T**
b. A variable name cannot be passed to a value parameter. **T**

c. If a C++ function does not use parameters, parentheses around the empty parameter list are still required. **T**
d. In C++, the names of the corresponding formal and actual parameters must be the same. **F**
e. Whenever the value of a reference parameter changes, the value of the actual parameter changes. **T**
f. In C++, function definitions can be nested; that is, the definition of one function can be enclosed in the body of another function. **F**
g. Using global variables in a program is a better programming style than using local variables, because extra variables can be avoided. **F**
h. In a program, global constants are as dangerous as global variables. **F**
i. The memory for a static variable remains allocated between function calls. **T**

Question No. 7

a. Explain the difference between an actual and a formal parameter.
b. Explain the difference between a value and a reference parameter.
c. Explain the difference between a local and a global variable.

Question No. 8

Write the definition of a void function that takes as input a decimal number and as output 3 times the value of the decimal number. Format your output to two decimal places.

Question No. 9

Write the definition of a void function that takes as input two decimal numbers. If the first number is nonzero, it outputs second number divided by the first number; otherwise, it outputs a message indicating that the second number cannot be divided by the first number because the first number is 0.

Question No. 10

Write the definition of a void function that takes as input two parameters of type int, say sum and testScore. The function updates the value of sum by adding the value of testScore. The new value of sum is reflected in the calling environment.

Question No. 11

What is the output of the following program?

```
#include <iostream>
using namespace std;

void find(int a, int& b, int& c,)

int main()
{
    int one, two, three;

    one = 5;
    two = 10;
    three = 15;

    find(one, two, three);

    cout << one << ", " << two << ", " << three << endl;

    find(two, one, three);

    cout << one << ", " << two << ", " << three << endl;

    find(three, two, one);

    cout << one << ", " << two << ", " << three << endl;

    find(two, three, one);

    cout << one << ", " << two << ", " << three << endl;
    return 0;
}


void find(int a, int& b, int& c)

{
        int temp;

        c = a + b;
        temp = a;
        a = b;
        b = 2 * temp;
}
```

5, 10, 15
20, 10, 15
25, 30, 15
45, 30. 60

Question No. 12

What is the output of the following program?

```
#include <iostream>
using namespace std;

int x;

void summer(int&, int);
void fall(int, int&);

int main()

{
        int intNum1 = 2;
        int intNum2 = 5;
        x = 6;

        summer(intNum1, intNum2);

        cout << intNum1 << " " << intNum2 << " " << x << endl;

        fall(intNum1, intNum2);

        cout << intNum1 << " " << intNum2 << " " << x << endl;

return 0;

}

void summer(int& a, int b)

{
        int intNum1;
        intNum1 = b + 12;
        a = 2 * b + 5;
        b = intNum1 + 4;
}


void fall(int u, int& v)

{
        int intNum2;
        intNum2= x;
        v = intNum2 * 4;
        x = u - v;
}
```

15, 5, 6
15, 24, -9

Question No. 13

Consider the following program:

```
#include <iostream>
#include <cmath>
#include <iomanip>
using namespace std;

void traceMe(double x, double y);

int main()

{
        double one, two;

        cout << "Enter two numbers: ";
        cin >> one >> two;
        cout << endl;

        traceMe(one, two);

        traceMe(two, one);

return 0;

}

void traceMe(double x, double y)

{
        double z;

        if (x != 0)
                z = sqrt(y) / x;
        else
        {
                cout << "Enter a nonzero number: ";
                cin >> x;
                cout << endl;

                z = floor(pow(y, x));

        }
        cout << fixed << showpoint << setprecision(2);
        cout << x << ", " << y << ", " << z << endl;
}
```

a. What is the output if the input is **3 625**? **3.00, 625.00 8.33**
625.00, 3.00, 0.00

b. What is the output if the input is **24 1024**?

24.00, 1024, 1.33
1024.00, 24.00, 0.00

c. What is the output if the input is **0 196**?

Enter a nonzero number: 2
2.00, 196.00, 38416.00
196.00, 0.00, 0.00

Question No. 14

Consider the following program. What is its exact output? Show the values of the variables after each line executes, as in Example 7-6.

```
#include <iostream>
using namespace std;

void funOne(int& a);

int main()

{
        int num1, num2;

        num1 = 10; num2 = 20;
        cout << "Line 3: In main: num1 = " << num1<< ", num2 = " << num2 << endl;

        funOne(num1);

        cout << "Line 5: In main after funOne: num1 = "<< num1 << ", num2 = " << num2;

return 0;

}

void funOne(int& a)

{
        int x = 12;   int z;

        z = a + x;
        cout << "Line 8: In funOne: a = " << a << ", x = " << x << ", and z = " << z << endl;
        x = x + 5;
        cout << "Line 10: In funOne: a = " << a << ", x = " << x << ", and z = " << z << endl;
        a = a + 8;
        cout << "Line 12: In funOne: a = " << a << ", x = " << x << ", and z = " << z << endl;
}
```

Line3: In main: num1=10, num2=20
Line8: In funone: a=10, x=12 and z=22
Line10: In funone: a=10, x=17 and z=22
Line12: In funone: a=18, x=17 and z=22
Line5: In main after funone: num1=18, num2=20

Question No. 15

What is the output of the following program?

```
#include <iostream>
using namespace std;
void tryMe(int& v);
int main()
{
	int x = 8;
	for (int count = 1; count < 5; count++)
		tryMe(x);
return 0;
}
void tryMe(int& v)
{
	static int num = 2;
	if (v % 2 == 0)
	{
		num++;
		v = v + 3;
	}
	else
	{
		num--;
		v = v + 5;
	}
	cout << v << ", " << num << endl;
}
```

11, 3
16, 2
19, 3
24, 2