Worksheet 1(C++ Function basic)

1. Define and explain the following terms
    a) Monolithic programming.
    b) Modular programming
    c) Module
    d) Factoring
    e) Structure chart or function tree
2. Compare Monolithic and modular programming based on their merit and demerit.
3. Some of the following pieces of code compile correctly but give unexpected results. Describe the behavior of the problems, if any, and fix them.
    a. Code to find the cube of a number

```
int x,y;
cout<<”Enter x:”;
cin>>x;
y=x*x*x;
cout<<"y=<<y";
```

    b. Code that calculates velocity and displays the result for a valid time input

```
/*assume that the function calculateV
is already defined*/
float T;
cin>>T;
if (T>10)
     cout<<”Invalid time: t>10s\n”;
else
     float V = calculateV(T);
     cout<<”V = ”<<V<<endl;
```

    c. Code to loop until the user decides to quit

```
int a, i=1;
do
{
    cout<<”Loop ”<<i++<<endl;
   cout<<”Enter 1 to loop again, ”<<”any other no. to exit: ”;
    cin>>a;
}while (a=1);
```

4. You are supposed to write a simple command line based matrix processor. It accepts a simple binary matrix expression, evaluates the expression and display the result. The possible commands are defined as follows:

Creating a matrix	A=(2 3 5: 6 7 8: 9 0 1: 4 2 0) // A is 4x3 matrix
Displaying a matrix	type the name of the matrix and press enter
Adding already created matrices	A + B or C=A+B
Subtracting already created matrices	A - B or C=A-B
Multiply already created matrices	A * B or C=A*B
Multiply already created matrices by scalar	2*A 0r C=2*A
Transpose of a matrix	A' or C=A'
Determinant	[A]

A single letter is used to name a matrix. The letter R is reserved as system name to hold the result of a matrix expression that is not assigned to another matrix (such as A + B). User can also use this matrix in their expression. Break down the problem into modules (functions) and draw the structure chart.

5. List and explain the different parts of a function
6. Identify the errors in the following function definition

```
fun(float x, int y)
{
    int r;
    …
    return r;
}
```

```
int fun(float x, int y)
{
    int r;
    …
    …
}
```

```
int fun(float x, y)
{
    int r;
    …
    return r;
}
```

```
void fun(float x, int y)
{
    float r;
    …
    return r;
}
```

7. The following programs generate compiler errors. Fix all the errors and show the outputs.

a.

```
1. #include <iostream.h>
2. int main() {
3.      int x=10, y;
4.      while (x>0) {
5.           y=0;
6.           for {int i=0; i<3; i++}
7.                y=y+i*x--;
8.           cout<<++y<<endl;
9.      }
10.  return 0;
11. }
```

b.

```
1. #include <iostream.h>
2. int main() {
3.          float Nat = 2.7e0F;
4.          cout<<"\'e\' squared = "<<nat*nat<<endl;
5.          return 0;
6. }
```

c.

```
1. #include <iostream.h>
2. int main() {
3.          const unsigned short LIMIT;
4.          LIMIT = 15;
5.          for (unsigned short t=1; t<=LIMIT; ++t)
6.          {
7.                    if (t%2 == 1 || (t > LIMIT/2 && t!=10))
8.                              continue;
9.                    cout<<t<<", ";
10.         }
11.         return 0;
12. }
```

8. For each of the functions in problem, give the complete definition of the function
9. For each of the function in problem write a complete program, by providing a main function that uses or call the functions. For c and d use only one main function that uses both functions.
10. Consider the following function definition

```
void nothing(){
    return;
}
```

```
int sum(int a, int b){
  return a+b;
}
```

Identify the correct calling. If not a valid call tell why you say that. Assume x, y and r are int type and z is float type. x, y and z contain 2, 4, 6.2 respectively. Whenever possible tell the change effected in r;

a. r=nothing();
b. sum(x,y);
c. r=sum(x);
d. r=sum(2,4);
e. z=sum(x,5);
f. nothing(x)
g. r=sum(2*x,3*y);
h. r=sum(sum(x,y), -y);
i. r=sum(sum(sum(x),y), -y);
j. r=sum(x, y, z);
k. r=sum(x,z)
l. nothing();

11. Tell the output of the following program

```
#include <iostream.h>
int x=0;
void printX ();
int main(){
  int x = 1;
 cout << x << endl;
 {
    cout << x << endl;
    cout << ::x << endl;
    int x = 2;
   cout << x << endl;
  {
    cout << x << endl;
    int x = 3;
    cout << x << endl;
    cout << ::x << endl;
  }
  cout << x << endl;
 }
 cout << x << endl;
 cout << ::x << endl;
 printX();
 return 0;
}
void printX(){
   cout<<x<<endl;
}
```

12. Refer problem 10;
    a. What is the significant of the statement on line number 3. Will the program compile if it is commented?
    b. For each definition of x tell the type of scope and the accessibility range in terms of line number
    c. What is the difference between file scope and program scope?