Tutorial - Week 10

Objectives: To practice with C++
- Reference function parameters
- Default function parameters
- Function overloading
- Function templates

1. **Will this code compile successfully? If your answer is NO, what is the problem and how will you fix it?**

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

void showVolume( int length, int width = 1, int height = 1  );

int main()
{
     showVolume( 2, 3, 4 );
     showVolume( 2, 3 );
     showVolume( 2 );
     showVolume( );

     return 0;
}

void showVolume(int length, int width = 1, int height = 1    )
{
     cout << "Volume of a box with \n"
         << "Length = " << length << ", Width = " << endl;
         << "and Height = " << height
         << " is  " << length*width*height<< endl;
     return;
}
```

2. Explain how these three function calls will be interpreted by C++ compiler.
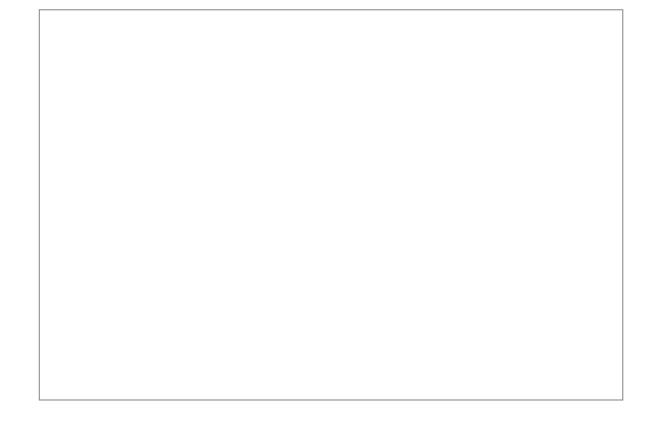
```
template <typename
T> T getMax( T p1, T
p2 )
{
  if (p1 < p2) return p2;
  else return p1;
}

double c, a=2.7,
b=6.9; c = getMax( a,
b );

char z, x='J',
y='A'; z = getMax(
x, y );

int ia = 7, ib = 2;
int v = getMax( ia, ib );
```

3. You need to implement a C++ function that takes two parameters of type unsigned char**. It increments them providing that both operations don't cause overflow. The function returns** true **if both values have been incremented or** false **otherwise. How can all values be passed back if only one value can be returned by a function?**

What is wrong with this function prototype?

```
bool incrValues( unsigned char& days=0, unsigned char& hours=0  );
```

4. **If variables a, b, and c are 504, 302.558, and −12.31, respectively, write a C++ program fragment that will display the line below. Do not display any quoted strings of blanks. Rather, adjust the field widths using setw to produce the blanks shown. (For clarity, a ▌ denotes a blank space.)**

   ▌▌504▌▌▌▌▌302.56▌▌▌▌−12.3

5. **write a program to sort three numbers in C++. Be sure to write a function called order to achieve this. Use two reference parameters in function order.**

6. **What two kinds of parameters does C++ allow functions and operators to have? What is the difference between these two kinds?**