Question 1: Explain the error, if any, in the following code: [1.5]

```
double max(double a, double b)      { return a>b ? a : b; }
double max(int a,int b)             { return a>b ? a : b; }
int main()                          {  cout << max(1,2.2); }
```

The arguments in the function call do not match either of the function definitions.

Question 2: **List** at least three things that the compiler will automatically provide for the following class [1.5]

```
class foo{ private: int a};
```

1. Default Constructor
2. Default Destructor
3. Default Copy Constructor

Question 3: What is the appropriate prototype for the copy constructor for a **`class`** called **Foo**? [1]

Foo(const Foo&);

Question 4: Write a class called **`Counter`** that contains two class data members, **`numBorn`** and **`numLiving`.** The value of **`numBorn`** should be equal to the number of objects of the class that have been created (even if they have been destroyed during the execution of the program). The value **of `numLiving`** should be equal to the total number of objects in existence currently (i.e, the objects that have been constructed but not yet destructed.) [4]

```
class counter{
   static int numLiving;
   static int numBorn;
   public:
    counter()
      {
         numLiving++;
          numBorn++;
      }
      ~counter()
      {
         numLiving--;
      }
};
int counter::numLiving = 0;
int counter::numBorn = 0;
```

Question 5: Modify the following class to write one constructor equivalent to the three constructors so that the output of a program using the Point class remains the same. [2]

```
class Point {
    int x, y;
public:
    Point():x(0),y(0){}
    Point (int xVal):x(xVal),y(0){}
    Point (int xVal,yVal):x(xVal),y(yVal) {}
    void display( );
};
```

```
Point(int xVal =0, int yVal = 0)
{
     x= xVal;
     y = yVal;
}
```

You can also use the initializer list notation.

Question 6: Consider the following piece of code. Which lines will give an error and why? [2]

```
class Car {
    double length;
    double weight;
public:
    void func(double,const double) const;
};

void Car :: func(doube newWeight, const double newLength) const {
    weight++;  ------→ ERROR because the function func() is declared constant
    newWeight += weight;
    length++; ------→ ERROR because the function func() is declared constant
    newLength += length; ------→ ERROR because newLength is passed as a constant
argument
}
```

Question 7: What would be the output of the following code fragment? Assume that **m** is at address 1000 whereas **n** is at address 1004. Write your answers in front of **cout** statements. [3]

```
int main(){
int m,n;
int& nn = n;
nn = 5;
m=10;
cout<<&n<<endl;  1004
cout<<&nn<<endl; 1004
nn++;
cout<<n<<endl;   6
nn =  m;
cout<<++nn<<endl;  11
cout<<&nn<<endl;   1004
cout<<&m<<endl;    1000
}
```