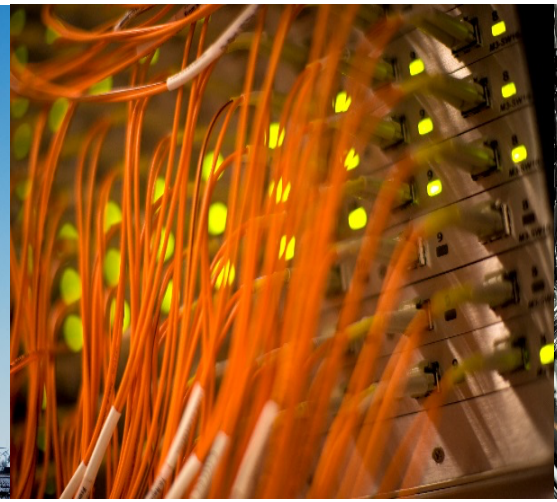
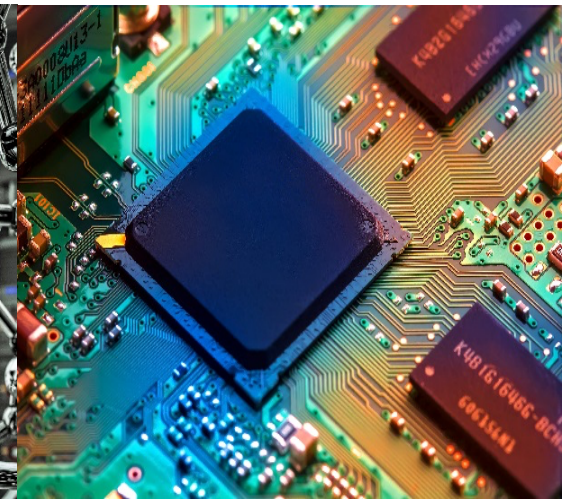
ECE 220 Computer Systems & Programming

Lecture 34 – C++ Examples

August 4, 2020



ILLINOIS
Electrical & Computer Engineering
GRAINGER COLLEGE OF ENGINEERING

- MP7 & MP8 due tomorrow
- Schedule your final exam with CBTF
- Final: 7pm on Friday, August 7th
- Conflict: 10:30am on Saturday, August 8th

Lecture 33 Review - Virtual Function

- **virtual functions** are member functions in the base class you expect to redefine in the derived classes
- derived class declares instances of that member function

```
class Shape{
   protected:
   int width, height;
   public:
   Shape(int a, int b) { width = a; height = b; }
   virtual int area() {
      cout << "Base class area unknown." << endl; return 0; }
};
class Rectangle : public Shape{
   public:
   Rectangle(int a, int b) : Shape(a,b){}
   int area() {
      cout << "Rectangle class area = " << width*height << endl;
      return width*height; }
};
```

Lecture 33 Review - Virtual Function Table (VTable)

- stores pointers to all virtual functions
- created for each class that uses virtual functions
- lookup during the function call

Function Template

```
//functions can have the same names (overload)
int sum(int a, int b){
        return a+b;
}

double sum(double a, double b){
        return a+b;
}


//define function with generic type instead!
template <class T>
T sum (T a, T b){
        return a+b;
}

int main(){
        cout << sum(5,7) << endl;
        cout << sum(1.5, 2.7) << endl;
}
```

Class Template

```
template <class T>
class mypair {
   T a, b;
   public:
   mypair (T first, T second)
      {a=first; b=second;}
   T getmax ();
};

template <class T>
T mypair<T>::getmax () {
   T retval;
   retval = a>b? a : b;
   return retval;
}
```

```
int main () {
    mypair <int> myobject (100, 75);
    cout << myobject.getmax();
    return 0;
}
```

Friend Function and Class

- Allows outside function/class to access a class' private and protected members
- "Friendship" is one-way