**CS Lecture 1-20**

**Quiz 3 Notes:**

int matches = 0;

for (int i = 0; i < count; i++)

{

If(name == students[i])

{

Cout << studentID[i] << endl;

Matches++;

}

}

if(matches == 0)

{

Cout <<”No students named “ << name << endl;

}

**Object Oriented Programming (OOP)**

* Centered around abjects
* Objects encapsulate data and functions together
* Examples: C++, C#, Java, Python, etc.

Vs.

**Procedural Programming**

* Focus on procedures or actions
* Examples: BASIC, C, Fortran, Pascal, etc.

**Class**

* Description of an object
  + Code that specifies:
    - 1) Attributes
    - 2) Functions
* Ex:

class Rectangle

{

Private:

Double length;

Double width;

String color;

public: //programmer can interact with

//setters and getter functions

void setLength(double);

void setWidth(double);

void setColor(string);

double getWidth() const; //protects from //modification

//const means functoin wll not modify data

double getLength() const;

string getColor() const;

double getArea() const;

};

**Access Specifiers**

* Private:
  + cannot be accessed directly
* Public
  + member functions can be accessed outside of the class
  + Provides the interface for Rectangle objects
* ::
  + Scope resolution operator
  + Separates class names and function name

**Types of functions**

* Accessors/getters:
  + Retrieve values (use const)
* Mutators/setters:
  + Change values

**Defining an instance of a class**

Rectangle box;

//Instantiation, created object called box

**Access an object’s members**

Box.setWidth(23.7);

Box.setLength(44.9);

Box.setColor(“red”);

Cout << box.getArea();

**Assignment 3 Notes**

**UML (Unified Modeling Language)**

* Class Diagram

|  |
| --- |
| Rectangle |
| * width:double * length:double * color:string |
| + setWidth(double):void  + setLength(double):void  + setColor(string):void  + getWidth() const:double  + getHeigth() const:double  + getColor() const:string  + getArea() const:double  + print() const:void |