Lecture 20

Pure Virtual Function

A pure virtual function (or abstract function) in C++ is a virtual function for which we don’t have implementation, we only declare it. We must override that function in the derived class. A pure virtual function is declared by assigning 0 in declaration.

```
 // An abstract class
 class Test {
     // Data members of class
 public:
     // Pure Virtual Function
     virtual void show() = 0;

     /* Other members */
 };
```

in above example, Test is an abstract class because it has a pure virtual function.

A complete example:
A pure virtual function is implemented by classes which are derived from a Abstract class. Following is a simple example to demonstrate the same.

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

 class Base
 {
    int x;

public:
    virtual void fun() = 0;
    int getX() { return x; }
};

// This class inherits from Base and implements fun()
class Derived: public Base
{
    int y;
public:
    void fun() { cout << "fun() called"; }
};

int main(void)
{
    Derived d;
    d.fun();
    return 0;
}
```

Output:

```
fun() called
```