# Project #4 Hints

```
/*************
* You can use this code as hints to do your assignment#4.  This code is very similar
* to assignment#4 specifications; please make sure to read assignment#4 specifications
* carefully  so you do not lose any points in your assignment.  Assignment#4 requires
* you to use pure virtual functions and to use command line argument for data file
* input.  Make sure to read Other grading requirements that are given in the
* assignment#4 specifications.

* Following is the data for the input file, proj4hints.data, used for these hints:
1,1,5,13
0,3,5
0,5,9
0,7,7
1,4,7,17
0,8,5
1,9,9,12
1,5,5,10
1,2,9,15
0,8,5

* The first column in the data file is the type of the fruit:
// Pineapple = 0
// Orange = 1

* If fruit is Pineapple than it has ID and Color its data.  If fruit is Orange it has
* ID, Color and nTotalSeeds its data.

* To do assignment#4, you must finish lesson #8 which is about run time polymorphism
* using pure virtual functions.  Make sure to complete lesson #8 before attepting to
* do assignment#4.

**************/
```

 

```
#include <iostream>
#include <fstream>
#include <iomanip>
#include <cstdlib>
#include <cstring>
using namespace std;

// This data structure should be a record in a input file
struct FruitFile {
      int Type;
      int ID;
      int Color;
      int nTotalSeeds;
};

enum ColorID {GREEN = 5, YELLOW = 7, BROWN = 9};
enum FruitType {PINEAPPLE,ORANGE};



class CFruit {
      public:
                  CFruit() { }
                  virtual void GetData(FruitFile &data) = 0;
                  virtual void ShowData() = 0;
                  const char *SetColorName(ColorID Color);
                  FruitType GetFruitType() { return m_FruitType; }
      protected:
                  FruitType m_FruitType;
                  int m_FruitID;
};

// Function: SetColorName
const char *CFruit::SetColorName(ColorID Color)
{
      switch (Color) {
            case GREEN:
                  return "Green";
                  break;
            case YELLOW:
                  return "Yellow";
                  break;
            case BROWN:
                  return "Brown";
                  break;
            default:
                  return "No Color";
                  break;
      } // end switch
}

class CPineapple : public CFruit {
      public:
                  CPineapple() : CFruit() {}
                  void GetData(FruitFile &data);
                  void ShowData();
      private:
                  ColorID m_ShellColor;
};
```

```
// Function: GetData
void CPineapple::GetData(FruitFile &data)
{
      // cast integer to enum
      m_ShellColor = ColorID(data.Color);

      // cast integer to enum
      m_FruitType = FruitType(data.Type);
      m_FruitID = data.ID;
}

void CPineapple::ShowData()
{
      cout << m_FruitType << setw(16) << m_FruitID << setw(20);
      cout << SetColorName(m_ShellColor) << "\n";
}

class COrange : public CFruit {
      public:
                  COrange() : CFruit() {}
                  void GetData(FruitFile &data);
                  void ShowData();
      private:
                  int m_TotalSeeds;
                  ColorID m_PeelColor;
};

void COrange::GetData(FruitFile &data)
{
      m_TotalSeeds = data.nTotalSeeds;

      // cast integer to enum
      m_PeelColor = ColorID(data.Color);

      // cast integer to enum
      m_FruitType = FruitType(data.Type);
      m_FruitID = data.ID;

}


void COrange::ShowData()
{
      cout << m_FruitType << setw(16) << m_FruitID;
      cout << setw(20) << SetColorName(m_PeelColor);
      cout << setw(13) << m_TotalSeeds << "\n";
}
```

 

```
// Pineapple = 0
// Orange = 1
int main(int argc, char *argv[])
{
      if (argc != 2) {
            cout << "Usage: PR <filename>\n";
            return 1;
      }

      ifstream Infile(argv[1]);
      if (!Infile) {
            cout << "Cannot open file\n";
            return 1;
      }

      char LineBuf[100];
      char d[] = ",";

      CFruit *pFruit[10];
      int i=0;
      while (Infile.getline(LineBuf, 100) && !Infile.eof()) {
            struct FruitFile data;

            data.Type = atoi (strtok(LineBuf, d));

            switch (data.Type) {
                  case PINEAPPLE:
                  // Create Pineapple Object
                        pFruit[i] = new CPineapple();
                        data.ID = atoi (strtok(NULL, d));
                        data.Color = atoi (strtok(NULL, d));
                        break;
                  case ORANGE:
                  // Create Orange Object
                        pFruit[i] = new COrange();
                        data.ID = atoi (strtok(NULL, d));
                        data.Color = atoi (strtok(NULL, d));
                        data.nTotalSeeds = atoi (strtok(NULL, d));
                        break;
                  default:
                        break;
            } // end switch

            // call appropriate function
            pFruit[i++]->GetData(data);
            memset(LineBuf,'\0',100);
      }

      cout << "Following are Pineapple values\n";
      cout << "\nFruit Type" << "\tFruit ID";
      cout << "\tShell Color" << "\n";
```

 

```
      for (int i = 0; i < 10; i++) {
            if (pFruit[i]->GetFruitType() == PINEAPPLE)
                  pFruit[i]->ShowData();
      }

      cout << "\n\nFollowing are Orange values\n";
      cout << "\nFruit Type" << "\tFruit ID";
      cout << "\tPeel Color" << "\tTotal Seeds" << "\n";

      for (int i = 0; i < 10; i++) {
            if (pFruit[i]->GetFruitType() == ORANGE)
                  pFruit[i]->ShowData();
      }

      for (int i = 0; i < 10; i++) {
            if (pFruit[i])
                  delete pFruit[i];// Delete appropriate object
      } // end for loop

      return 0;
}
```

**OUTPUT:**

```
Following are Pineapple values

Fruit Type   Fruit ID     Shell Color
0            3            Green
0            5            Brown
0            7            Yellow
0            8            Green
0            8            Green


Following are Orange values

Fruit Type   Fruit ID Peel Color   Total Seeds
1            1            Green       13
1            4            Yellow      17
1            9            Brown       12
1            5            Green       10
1            2            Brown       15
```