CMP-142 Object Oriented Programming
BS Fall 2014
Home Task 10

Issue Date: 12-May-2015
End Date: 13-May-2015

Objective:

- It will help getting a grip on array of object and association relationship.

Task-0:

Here is detail task that was given in your mid term exam.
Our objective is to make an ADT 'VotingMachine' which will be able to compute election results for candidates contesting within one constituency.

The class "Candidate" is used to store the information about a candidate who is contesting in election. The class "VotingMachine" keeps records of all the candidates contesting in election and is used to cast votes for a particular candidate (an object of class "Candidate").

```
class Candidate
{
private:

        CString name;                //name of candidate

        CString elecSymbol;          //election symbol of candidate

        int votes;                   //number of votes casted to candidate

public:
```

1. Candidate(const CString nam);
   receives name of candidate and election symbol

2. CString getName() const;

3. void setName(const CString nam);

4. CString getElecSymbol() const;

5. void setElecSymbol(const CString nam);

6. int getVotes() const;

7. void castVote();
   increment of 1 in the vote count of candidate

```
};

class VotingMachine
{
private:

        int numOfCandidates;                 // number of candidate objects

        Candidate * candidateList;           // points to an array of candidate objects

public:
```

1. VotingMachine (const Candidate * P, const int N);
   Parameter "N" tells the number of candidates, so data member "candidateList" *will point to an array of objects of type Candidate of size "N".*
   Parameter "p" receives the array of candidate objects.

2. int castVote(const CString &);
   It invokes member function the "castVote()" for the Candidate object whose election symbol is recieved.

3. int candidateReport(const CString &) const;
   It returns the number of votes casted so far for the candidate whose election symbol is received.

4. Candidate* electionResult() const;
   It return an array of candidates having 3 objects with winner at index 0 and runner-up at index 1and 3rd position at index 2.

};

Note:

- You are free to add private/public members in 'CString' and 'Candidate' class if needed but that need has to be imperative☺ otherwise a penalty will be put on.

Task-1:

Suppose you have two classes 'Apple' and 'Basket' as follows:

```
class Apple
{
public:
    int whichColorBasket()
    {
            // gets the color of the
            //basket the apple is in...?
    }
};
class Basket
{
        int basketColor;
        Apple * fruits[10];
public:
    Basket(int col)
    {
        basketColor = col;
        for (int i=0; i<10; i++)
        {
                fruits[i] = new Apple(this);
        }
    }
        int getColor()
        {
                return color;
        }
};
```

In main/driver program, we have created many baskets which in turn generate 10 apples in each basket. So basically, main/driver will have many objects of Baskets; and baskets will have many objects of Apples. If we have a function in apple that needs to know the color of my basket (the basket in which apple is located), how can we go about finding the color of the basket?
Note: You are free to add any member function or data member in class 'Apple' and class 'Basket' if needed; in order to achieve the required feature.