**Free Response Question**

1 Assume that information about candidates in a class election is stored using the Candidate and CandidateList classes below:

```
public class Candidate
{
        private String myName;
        private int myNumVotes;
        private double myVotePercent;
        // constructor:  myVoteCount is initialized to 0.  Actual value is set later.
        public Candidate (String name, int numVotes)
        {       /*  implementation not shown */  }
        //  Set myVotePercent equal to votePercent
        //  Precondition: votePercent is a real  number between 0.0 and 100.0
        public void setVotePercent ( double votePercent )
        {  myVotePercent = votePercent;  }
        //  accessors
        public String getName ( )
        {  return myName;  }
        public int getNumVotes ( )
        {  return myNumVotes;  }
        public double getVotePercent ( )
        {  return myVotePercent; }
}
public class CandidateList
{
        private Candidate [ ] myCList;
        private int size;
        //  constructor:  Reads name and number of votes for all candidates into myCList
        public CandidatesList ( )
        {  /*  implementation not shown */  }
        //  Precondition:       myCList contains Candidate objects.
        //  Postcondition:      The vote percent for each Candidate has been calculated and updated.
        public void computeVotePercents()
        {  /*  to be implemented in part (a) */  }
        //  Precondition:       myCList contains complete information about all candidates, including their
        //                      updated vote percents.  Each vote percent is a real number
        //                      between 0.0 and 100.0.
        //  Postcondition:      Returns a list of viable candidates, namely those candidates who got at
        //                      least 10 percent of the vote.
        public Candidate [ ] getViableList()
        {  /*  to be implemented in part (b) */  }
        //  Precondition:       myCList contains complete information about all candidates, including
        //                      their updated vote percents.
        //  Postcondition:      The names of viable candidates only have been printed, one per line,
        //                      followed by that candidate's vote percent.
        public Candidate [ ] printViable()
        {  /*  to be implemented in part (c) */  }  }
```

a) Write the implementation of the computeVotePercents method of the CandidateList class. The computeVotePercents method should fill in the vote percent for each Candidate in myCList. A candidate's vote percent is computed by dividing the number of votes for that candidate by the total number of votes cast for all candidates and then multiplying by 100.

In writing computeVotePercents, you must use any of the accessible methods in the Candidate and CandidateList classes.

Complete method computeVotePercents below:

```
// Precondition:    myCList contains Candidate objects.
// Postcondition:   The vote percent for each Candidate has been calculated and updated.
public void computeVotePercents( )
```

b) Write the implementation of the getViableList method of the CandidateList class. The getViableList method should examine the elements in myCList and create an array of viable candidates only. A viable candidate is one who received at least 10 percent of the vote.

In writing getViableList, you must use any of the accessible methods in the Candidate and CandidateList classes.

Complete method getViableList below:

```
// Precondition:    myCList contains complete information about all candidates, including
//                  their updated vote percents. Each vote percent is a real number
//                  betwen 0.0 and 100.0.
// Postcondition:   Returns a list of viable candidates, namely those candidates who got at
//                  least 10 percent of the vote.
public Candidate [ ] getViableList( )
```

c) Write the implementation of the printViable method of the CandidateList class. The printViable method should list the names and vote percents of viable candidates only, one per line. Sample output:

```
Chris Arsenault  42.4278542118662
Anton Kriksunov 15.8023902117245
Lila Fontes  29.764689012392
```

In writing printViable you must call the getViableList method specified in part (b), and use the list returned. Assume that getViableList works as specified regardless of what you wrote in part (b).

Complete method printViable below:

```
// Precondition:    myCList contains complete information about all candidates, including
//                  their updated vote percents.
// Postcondition:   The names of viable candidates only have been printed, one per line,
//                  followed by that candidate's vote percent.
public void printViable( )
```