

Suggested Solutions to Free-Response Questions

Note: There are many correct variations of these solutions.

Question 1

(a)

```
private int numFirstVotes(String candidate,
                           List<String> candidateList)
{
    int numVotes = 0;

    for (Ballot voterBallot : ballotList)
    {
        String first = voterBallot.firstChoiceFrom(candidateList);

        if (candidate.equals(first))
            numVotes++;
    }
    return numVotes;
}
```

(b)

```
public List<String> candidatesWithFewest(
    List<String> candidateList)
{
    int[] votes = new int[candidateList.size()];
    int minVotes = ballotList.size();

    for (int c = 0; c < candidateList.size(); c++)
    {
        String candidate = candidateList.get(c);
        votes[c] = numFirstVotes(candidate, candidateList);

        if (votes[c] < minVotes)
            minVotes = votes[c];
    }

    List<String> result = new ArrayList<String>();
    for (int c = 0; c < candidateList.size(); c++)
    {
        if (votes[c] == minVotes)
            result.add(candidateList.get(c));
    }

    return result;
}
```

(b) Alternate solution

```

public List<String> candidatesWithFewest(
    List<String> candidateList)
{
    List<String> result = new ArrayList<String>();
    int minVotes = ballotList.size() + 1;

    for (int c = 0; c < candidateList.size();c++)
    {
        String candidate = candidateList.get(c);
        int thisVotes = numFirstVotes(candidate, candidateList);

        if (thisVotes < minVotes)
        {
            minVotes = thisVotes;
            result = new ArrayList<String>();
        }
        if (thisVotes == minVotes)
            result.add(candidateList.get(c));
    }

    return result;
}

```

Question 2

(a)

```

public void insert(String str, int index)
{
    myLine = myLine.substring(0, index) + str
            + myLine.substring(index);
}

```

(b)

```

public void delete(String str)
{
    int index = myLine.indexOf(str);

    if (index != -1)
    {
        myLine = myLine.substring(0, index)
                + myLine.substring(index + str.length());
    }
}

```

Sample Questions for **Computer Science A**

(c)

```
public void deleteAll(String str)
{
    while (myLine.indexOf(str) != -1)
    {
        delete(str);
    }
}
```

Question 3

(a)

```
public class SavingsAccount extends Account
{
    private double intrRate; // annual interest rate for
                            // this account

    public SavingsAccount(int idNumber, double balance,
                           double rate)
    {
        super(idNumber, balance);
        intrRate = rate;
    }

    public double monthlyInterest()
    {
        return (currentBalance() * (intrRate / 12.0));
    }

    public void withdraw(double amount)
    {
        decreaseBalance(amount);
    }
}
```

(b)

```

public class SpecialCheckingAccount extends CheckingAccount
{
    private double minBalance;
    private double intRate;

    public SpecialCheckingAccount(int idNumber,
                                   double startBal,
                                   double chkCharge,
                                   double minBal, double rate)
    {
        super(idNumber, startBal, chkCharge);
        minBalance = minBal;
        intRate = rate;
    }

    public void clearCheck(double amount)
    {
        if (currentBalance() >= minBalance)
            decreaseBalance(amount);
        else
            super.clearCheck(amount);
    }

    public double monthlyInterest()
    {
        if (currentBalance() >= minBalance)
            return (currentBalance() * (intRate / 12.0));
        else
            return super.monthlyInterest();
    }
}

```

(c)

```

public void postMonthlyInterest()
{
    double interest;

    for (Account acct : accounts)
    {
        interest = acct.monthlyInterest();
        acct.deposit(interest);
    }
}

```