CMPSCI 187, Spring 2015 Discussion #13:
Hash Functions: Response Sheet

Marc Liberatore and John Ridgway

27 April 2015

Names: _____

Each *pair* of students should hand in one response sheet.

Question 1: Here is a possible hash function. Does it meet the absolute requirement? Does it have the desirable characteristic?

```
1  public int hashCode() {
2    return 1;
3  }
```

Question 2: For this question assume that we are writing a `hashCode()` method for a class similar to the `String` class. Its sole attribute `characters` is an array of **char** that contains the characters making up the string.

One possible hash function would be to simply add the characters together, as follows:

```
1  public hashCode() {
2    int result = 0;
3    for (char c : characters) {
4      result += c;
5    }
6    return result;
7  }
```

(and yes, this works; characters can be added and subtracted and converted to integers).

State a potential problem with this. *Hint: think about strings with the same letters in a different order.*

Question 3: Considering the method in the previous question: Do you think it would be a good idea to only consider the first five characters so as to reduce the amount of computation?

Question 4: As before, assume that we are writing a `hashCode()` method for a class similar to the `String` class. Suppose we know that the the the input to this string will always be a URL representing a web site served by HTTP, located on the server `www.cs.umass.edu`. How might you improve the runtime of the previous hash function?

Question 5: As before, assume that we are writing a `hashCode()` method for a class similar to the `String` class. Further suppose that we've initialized a `Random` object as an attribute named `random` in this class.

```
1  public hashCode() {
2    int result = random.nextInt();
3    for (char c : characters) {
4      result = 31 * result + c;
5    }
6    return result;
7  }
```

Is this function more or less likely to return a unique value for each input than our previous attempt? And, is it a valid implementation of `hashCode`? Why or why not?