CS100M: Section Exercises for Nov 15-16

1 Bag of Dice

Assume you have an implementation of the Dice class from lecture. The class has the following public methods:

- public Dice(int numSides)
- public void roll()
- public int getTop()
- public int getSides()
- public void setTop(int faceValue)

In some games, several dice are rolled at once, which leads to interesting non-uniform probability distributions. So, now we will create a bag of dice, a class that holds several dice and has a method to roll all of them simultaneously.

```
class BagOfDice {
  // this array will hold all dice
  private Dice[] dice;

  /** Create a new bag of dice with given number of dice.
  * The number of sides will be 6 for all new dice. */
  public BagOfDice(int numDice) {

  }

  /** Create a new bag of dice with given numbers of sides.
  * The number of dice should be equal to the array length. */
  public BagOfDice(int[] numSides) {

  }

  /** returns the array of Dice objects being used by this class */
  public Dice[] getDice() {

  }

  /** Roll dice and return the sum of numbers on top. */
  public int rollAllDice() {

  }
}
```

You can download the DiceChecker.java file and run it to check if your code works.

2 Polynomials

In this part, we will create a class that represents a polynomial as an array of coefficients.

```
class Polynomial {
  // the array of coefficients
  // coeffs[i] is the coefficient next to x^i
  protected double[] coeffs;

  /** Create a new polynomial with the given coefficients. */
  public Polynomial(double[] coefficients) {

  }

  /** The degree of a polynomial is the degree of the
  * highest nonzero term. */
  public int getDegree() {

  }

  /** Compare this polynomial to a given one. */
  public boolean equals(Polynomial p) {

  }

  /* evaluates the polynomial with input x */
  public double evaluate(double x) {

  }
}
```

You can download PolynomialChecker.java and run it to check your code.