

CS-141 Basic Class & Object Problems

1. Create a `Die` class that has the items below. Note, none of the methods or variables should be static. Compile the class as you add methods to make sure your syntax is correct before implementing the next method. You might even start problem 2 (below) to test the methods as you implement them. It is best to compile and test as much as possible as you add complexity to your code.
 - a. **Instance Variables:** Create 2 instance variables – one for the number of sides and one for the value of the most recent roll.
 - b. **Accessor Methods:** For each instance variable, there should be a “setter” and “getter” method (i.e. `getSides`, `setSides`, `getRoll`, `setRoll`). Note, you probably don’t really want a `setRoll` since this will allow you the cheat!
 - c. **toString method:** create a `toString` method with signature:

```
public String toString()
```
 - d. **Constructors:** create 2 constructors
 - i. One that takes no parameters (default) and sets the number of sides to 6
 - ii. One that takes a single parameter representing the number of sides
 - e. **Actions (Instance Methods):** A `roll` method that
 - i. Rolls a die (as we have done before with `Math.random()`)
 - ii. Sets the instance variable to the value of the roll.
 - iii. Returns the value of the roll
2. Create a second class called `Game` that contains a `main` method. In `main`, create several `Die` objects to test each of the methods above.
3. Normally, very little is done in the `main` method. In `Game`, create a non-static method called `play` that creates and rolls two `Die` objects and prints the result (you can probably copy some of the code from problem 2 by). In `main`, remove your previous code, and create a `Game` object and call the `play` method on that object.
4. Try writing classes and creating objects for the following (pick one that interests you)
 - a. A `Card`
 - b. A `Deck of Cards` (contains an array of `Card` objects)
 - c. A `Student` (e.g. with name, grades, address, etc)
 - d. `Song` (name, artist, etc)
 - e. `Movie` (name, producer, studio, lead actors, description, ...)
 - f. `Car` (model, brand, year, mpg, etc)

When implementing each class, consider adding code for each of the categories below. Copy the headings into your Java file and fill in. You probably won't need all of these categories.

```
// Private Member Instance variables *****  
// Private Member Class (static) Variables *****  
// Constructors *****  
// Mutators and Accessors *****  
// Additional Public Methods *****  
// Public Class (static) Methods *****  
// toString Method *****  
// Private Methods *****  
// Private Class (Static) Methods *****
```