# Of Dice & Dragons, Part 1: Objects

Due: Monday, February 4th

**Objective**: To become accustomed in defining classes and creating objects. Although the scope of this assignment is relatively small in comparison to previous assignments, what will be learned here will serve towards the development of a text-based role-playing game (RPG).

**Description**: As discussed in previous classes, real-world objects may be described in the context of programming languages. Such languages are known as object-oriented programming (OOP) languages, which is one of many programming paradigms that exist. Both real-world objects and software objects contain state (values that describe them at a point in time) and behaviors (actions that the object can do). An important aspect of utilizing objects and OOP languages in general is the concept of *abstraction.* The main goal of abstraction is that the development environment of the language (in the case of Python, IDLE is the development environment) handles the complexity of many tasks all the while hiding the unnecessary details to the programmer so that they may focus on implementing more complex logic without having to fully understand what is going on behind the scenes. This may not seem like a very specific concept (and that's because the more general it is, the more power it has over many things), but there are plenty of examples of abstraction in the real world. For example, an average person doesn't need to understand the complexities and difficulties of electrical engineering to know how to use a microwave oven, right? It's a similar concept when it comes to making use of objects. An average person doesn't need to understand the complexities and difficulties of computer science to be able to use a piece of software. Thus, this assignment seeks to explore more in-depth how the concept of abstraction will become useful in representing real-world concepts as objects in Python.

**Problems**

1) Create a **Car** class. All cars have a make, model, year, and transmission type. If the car has a transmission type "M" for manual, create a variable for the number of gears the car has, which can be done by defining the method **gears()**. In addition, create a list of cars and print out each car's information on separate lines. Remember that the special function **__str__()** may be of help in printing out the car's information.

2) Create a **Student** class. In many school systems, students are kept track of by a unique student ID number, the grade level that they currently are in, expected graduation year, and their GPA. Create a list of students and print out each student's information on separate lines.

3) Create a **Die** (the singular version of the plural noun dice) class. Although the most common type of die known to many is the six-sided die, there are in fact a wide range of multiple dice with a different number of sides. For example, in the game of

Dungeons & Dragons, the most well-known die is twenty-sided, known as a d20. Use the number of sides a die may have as the only argument for your **__init__()** function. Include a method **roll()** that returns an integer that is within the possible range of the die. For example, for a six-sided die the expectation will be that the face value of it after it has been rolled will be within 1 and 6. Test the **roll()** method by first creating a list that will contain as many elements as there are sides, remembering to initialize every cell to 0, then calling the method in a for loop, keeping track of the rolled numbers within the list. Loop through the list once, printing out the overall results of all the rolls. In other words, suppose a six-sided die is created and tested. Thus, the list must have a capacity of 6 elements, and if the resulting roll for one of the loop iterations is 3, then the value in the third cell of the list will be incremented by 1.

## Questions & Discussion

1) Think about how these classes and objects could be used in incorporating real-world concepts into the AP Create projects. How can this be an example of abstraction?
2) The next step towards this project is the definition of some basic character classes. Begin to think on how this can be achieved via a basic **Character** class definition, which can act as a template for more diversified and/or specializes classes.

## Philosophy on the Creation of this Assignment

Personally, I was stuck on a bit of a rut when it came to brainstorming for new assignment ideas assignments as I wanted something that I felt would not only be beneficial to you, but also something fun. I was discussing with some friends on how this could be approached, and we got into a conversation on text-based games and how sometimes even a simple concept such as that can still carry with it its complexities, especially regarding its development. At that moment I was struck with a blossoming idea that now span perhaps two or three more future assignments (i.e. a total of three or four parts) that will allow many of you to obtain the much-needed skills of developing a software project. My hope is that these skills will be carried over and used in the development of your Create projects.

## Final Notes

As always, please direct any questions or concerns regarding not only this assignment, but also your Create projects to my email at diazg.gustavoa@knights.ucf.edu. You may expect a response within a 24-hour window, so please don't hesitate to send me an email.

In addition, I would like to thank the previous Colonial High School AP Computer Science Principles volunteer Kevin Negy for providing a good basis for this assignment on objects, and to my friends Alexander Mateo and Joshua Romero for the idea on this project.