05.05 Virtual Lecture Notes (Part 1)

Java programs can be written in several styles, which can be categorized as **`main()`** method, procedural, and object-oriented design programming.

The **`PlanetV2`** class uses procedural programming style. The **`PlanetV3`** class illustrates basic object-oriented programming in which a default constructor is used to create an object named **`space`** of type **`PlanetV3`**. The default constructor is located immediately below the class definition and uses the class name.

```
public PlanetV3()
{
}
```

It may seem odd that this simple statement, which contains no code between the curly braces, could really hold the key to object-oriented programming!

Next, turn your attention to the **`PlanetV3`** class's two methods: **`calcRadius()`** and **`printPlanet()`**. Notice that there is no longer a need for the **`static`** modifier that was used in the **`PlanetV2`** class (which used procedural programming style).

Some important modifications have been made in the **`main()`** method, which identify the object-oriented nature of the **`PlanetV3`** class. The following statement is a significant new addition to the program, but should look familiar because of the keyword **`new`**.

```
PlanetV3 space = new PlanetV3();
```

This is the same format previously used to construct **`PrintWriter`** and **`Scanner`** objects and is the standard format for invoking the constructor of a class to create an object. It consists of two parts separated by the equal sign.

The code on the left of the equal sign declares that there will be a new object named **`space`** of type **`PlanetV3`**, the right side invokes the default constructor to actually create the new object. This statement can also be written in two lines, as shown below:

```
PlanetV3 space;
space = new PlanetV3();
```

Once the **space** object is constructed, the methods of the **PlanetV3** class can be invoked to perform their calculations. Invoking a method is accomplished using dot notation to call the method on the object, as illustrated in the following statements:

```
//call methods
radius = space.calcRadius(diam);
space.printPlanet(name, radius);
```

How does this differ from invoking static methods? Refer to the **PlanetV2** class and you will notice that naming the object is not required, but everything else is the same.

When these statements are executed, flow of control jumps to the appropriate method and value of each variable listed as an argument is passed to the corresponding variable in the method's parameter list. The method then performs its task and returns an answer to be assigned to the variable on the left side of the equal sign.

To make the transition from procedural programs with static methods to object-oriented programs that explicitly construct an object requires four modifications:

1. Include a statement that invokes the constructor to create a **new** object.
2. Write the default constructor.
3. Write method headers, but do not include the **static** modifier.
4. Write statements in the **main()** program using dot notation to call methods for the object constructed.

Print