Virtual Lecture Notes

Let us take a look at the **abstract class `BasicJean`** first.

- Download the BasicJean.java file to your Unit 5 Lessons folder.
- Study the code carefully before you continue

Ms. Mills wants us to make a pair of beaded jeans. We are going to accomplish this by adding a number of beads in a particular pattern to the jeans. For programming purposes, this means we will create a **`BeadedJean`** class where we add an int **`numBeads`** to indicate the number of beads we will need and a string **`pattern`** that will hold the name of the pattern that will be made by the beads.

We start off by extending the **`basicJean`** class.

```
public class BeadedJean extends BasicJean
```

Now, for our exercise, we are going to assume that **`BeadedJean`** will be given a number of beads to go along with the number of pockets at the start. Since in the abstract class, design is an abstract method, we must now define it. To keep it simple for the purpose of this demo, design will use the number of beads to determine the pattern. If the number of beads is 10, the pattern will be set to circle. If the number of beads is 20, then the pattern will be set to heart. If the number of beads is 30, then the pattern will be set to spiral. And lastly, if the number of beads is anything else, the pattern will be random.

- Download the BeadedJean.java file to the current project folder and open it.
- Make sure you understand it before you continue

Notice that there is also a **`toString()`** method added, to say how many beads are used and in what pattern.

That is all there is to an abstract class. If we wanted to do so, we could make more types of jeans and just simply define a different design method for each one.

- Download the Tester.java file to the current project folder and open it.
- Run it to see the **`BeadedJean`** class in action.

Make sure you understand how the demo programs work before you proceed.