4. Class Use

Outline

- Operator new
- Operators and Operations
- Using operator new
- Constructors
- Saving new objects
- New objects and methods
Creating Objects

- So far, we’ve seen **predefined** objects
  - System.out which is a PrintStream object
  - String literals: “”, “Rah!”
- Invoked methods that have created objects for us
  - String.toUpperCase() creates a new string object and returns its reference

```
string VaTechCaps = "VaTech".toUpperCase();
```

- In general, use operator **new** to create a new object

Remember:
Declare references, create objects.
- Operation – an action that results in a value
- Operator – a symbol or keyword that represents an operation
- Example:
  - Operator: `new`
  - Operation: creates and returns reference to object

\[
\text{returned reference} \quad \text{new object}
\]
Using Operator new

- To create an object we must tell the `new` operator what type of object to create, and indicate how object is to be initialized

- Example:
  ```java
  new String();  // create new String object
  ```

- Use `constructors` of a class to indicate what type and specify initialization
  - `new` automatically calls the constructor!
Constructors

- A class may have many constructors, but all have the same name as the class – differ by arguments
  - `String()` – creates an empty string `{""}`
  - `String(“a string”)` – creates a string with literal
- Responsible for initializing an object
  - Constructors ensure that no object is ever created in an un-initialized state
- Constructors are usually overloaded

Remember what ‘overloaded’ means?
If we want to use an object past the statement in which it is created, we must use reference variables.

```java
String message;
message = new String("Your lights are on");
```

May need object only in current statement

```java
message = (new String("run! ")).toUpperCase();
```

This object is not needed beyond here

```
message
```

```
"Your lights are on 
```

```
reference returned by new
```

```
"run! 
```

```
reference returned by toUpperCase()
```

```
"RUN! 
```
New Objects and Methods

- Some methods return references to objects they create.
- The String methods like toUpperCase(), toLowerCase(), trim(), concat(), substring() are examples.
- When using these methods we still may need to store the references, but do not need to create objects ourselves, (i.e. the invoked methods create the new objects).

```java
string VT = " VaTech ".trim();
```

```
reference sent to trim()  \[ " VaTech " \]
```

```
string literal object created by Java compiler
```

```
VT    \[ "VaTech" \]  new string object created by trim()
```

```
reference sent to trim()  \[ " VaTech " \]
```

```
string literal object created by Java compiler
```

```
VT    \[ "VaTech" \]  new string object created by trim()
```