Content Objective: Students will implement their own classes for Java applications.

|  |  |
| --- | --- |
| **On the Tech Horizon (10pts.)**  **link to a tech/coding related article or journal no more than one month old (no blogs or reddit clones see below)** | |
| URL: |  |
| Reaction/Commentary: |  |

|  |  |
| --- | --- |
| **Tech Terms and History (20pts.)**  **vocabulary from BJ p.81-p.130 and The Information Chapter 2 (definition/commentary/significance in your words)** | |
| Instance vs. Local Variable | Local variables are declared inside of methods, instance variables are where objects store variables. (in the instance of the class) |
| Access Specifier | Declares what can access a variable or class |
| Private, Public, Protected | Public- any class can access it; Private-Only things that are able to access methods and variables are able to; Protected-the class itself, classes in the same package and subclasses can access |
| Member Method | Methods inside of a specific class |
| Encapsulation | Having hidden implementations but being publicly published |
| Public Interface | The information of a class given to people. Operations anyone can use to manipulate the class. |
| Private Implementation | Hiding or making variables only accessible by methods in the class |
| Constructor | The blueprint used to initialize different instantiations of an object |
| Parameter vs. Argument | BJ p.90 |
| Unit Test | BJ p.102 |
| Garbage Collector | BJ p.107 |
| Local Variable | BJ p.107 |
| this | BJ p.110 |
| Syllabary | TI p.33 |
| Memory was aided by? | TI p.34 |
| Abstract Thought | TI p. 35 |
| Classifying/Categorization | TI p.36 |
| Abstraction to Logic | TI p.37-38 |
| “To go by your words…” | TI p.39 |
| Horses and Classes | TI p.40-41 |
| Paradox and Mathematics | TI p. 41 |
| What is base 60? | TI p. 43 |
| Early Algorithm | TI p.45 |
| Algorithm and Procedure | TI p.46 |
| Thinking and Writing | TI p.47 |

|  |  |
| --- | --- |
| **Code Snippets (30pts.)**  **only submit snippets or classes no full programs required (test and run in IDE, then copy/paste applicable code frag)** | |
| R3.16 & R3.17 | BJ p.102-3 for checklists |
| R3.22 & R3.24 | BJ p.112-115 |
| E3.3-E3.8 | BJ p.93-103 |
| E3.12 | BJ p.125 |

|  |  |
| --- | --- |
| **Code Challenge (30pts.)**  **full functioning application sent in a Zip file to Canvas** | |
| Using programming projects from chapter 3 (BJ p.127-128), students may choose **one** of the following challenges P3.4, P3.5, P3.7, P3.8 or P3.9. As a part of this assignment, students will need to find a partner to independently write their own unit test for the class that is implemented. You may use Canvas to send a zip file to another student. The final submission will need to include all applicable Java source code files and a unit test file from a peer (containing their name in a comment or within the file name), compressed into one .zip file for grading. | |
| Zip File Contains: |  |
| Notes: |  |

|  |  |
| --- | --- |
| **Badge Progress (10pts.)**  **building your coding profile: Java coding training site to earn badges (recommended site** [**http://coderbyte.com**](http://coderbyte.com) **)** | |
| Screenshot/URL: |  |
| Notes/Issues: |  |

|  |  |
| --- | --- |
| **Notes**  **your notes** | |
| Notes: |  |