Code Review – Keith Fraser

Week 4

|  |  |
| --- | --- |
| **Code Formatting** | **Comments** |
| Indentation:   * Use of white space * Start and end point of code block easily identifiable | Good use of white space  Start and end easily identifiable |
| Naming Conventions:   * Meaningful names for classes, methods and variables   + Intention revealing names   + One word per concept * Use of case   + Variables = lowerCamel   + Classes = CamelCase   + Methods = lowerCamel   + Functions = lowerCamel | All meets the standard  Everything follows case structure |

|  |  |
| --- | --- |
| **Best Practices** | **Comments** |
| Comments:   * Class/files headers * Methods * Why, not what? | One in the starting class, others don’t have any  Methods, however, are commented. |
| No hard coding | No hard coding |

|  |  |
| --- | --- |
| **Non Functionality** | **Comments** |
| Maintainability:   * Readability   + Class: Variables, Constructor, Methods   + Logical order of methods * Understandability * Consistancy | Good readability and maintainability |
| Reusability   * Use of classes * Reusability of functionality * Avoid repetition of code – DRY (Do Not Repeat Yourself) * Functions should be small * Functions should do one thing * Classes should be small | Code is overall very reusable |
| Easy to test | Easy to test |

|  |  |
| --- | --- |
| **Fundamentals** |  |
| Use of frameworks (ie java.util.\*) | No use of frameworks |
| Use of encapsulation   * Getters and Setters | No getters or setters |
| Use of constructor overloading as appropriate | No constructor overloading |
| Similar values grouped (Enum) | No need enum |
| Input validation implemented | No need for input |
| Use try-catch loop for user input to program (Exception Handling) | Try-catch loop used in updating clock every second |

|  |  |
| --- | --- |
| **Functionality** | **Comments** |
| Does it meet the requirements? | Meets the requirements, has added seconds but hasn’t implemented 12-hour display. |
| Does it work? | It does work |