**Code Review Checklist**

|  |  |  |  |
| --- | --- | --- | --- |
| Student | Hristina Koleva | Date | 03.07.2013 |
| Program | Personal Software Process – PSP | Program # | 6 |
| Instructor | Valentina Ivanova | Language | C# |

|  |  |
| --- | --- |
| Purpose | To guide you in conducting an effective code review |
| General | * Review the entire program for each checklist category; do not attempt to review for more than one category at a time! * As you complete each review step, check off that item in the box at the right. * Complete the checklist for one program or program unit before reviewing the next. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Defect Log # | Category | * Criteria | Original Defect | Result |
|  | Function | * Validate correct indexing in for loops. Check for argument out of range exceptions and boundary conditions | 1 | ok |
|  | Function | * When calculating sums do not forget the incrementing assignment | 2 | ok |
|  | Data | * Validate input data is accessible. Handle cases when input data cannot be accessed | 3 | ok |
|  | Data | * Validate correctness of input data. Handle incorrect input data – wrong or missing data. | 4,7,6 | ok |
|  | Interface | * Validate user input. Handle incorrect input. Use proper user messages. | 5 | ok |
|  | Function | Correct loop handling. Make sure that loops are exited when required and endless loops are not introduced. Add return statements at their proper places. | 8,10,18 | ok |
| 36, 39 | Assignment | Make sure all variables are initialized properly when needed. Proper use of properties and fields. | 11,21,17 | ok |
|  | Assignment | Check conditional expressions – make sure conditional expressions return required result and boolean variables are assigned with required values before they are used. | 15 | ok |
| 41 | Function | Make sure all methods are called in correct order and when needed. | 24, 25, 26 | ok |
|  | Function | Check proper use of stream. Validate streams are not locked and can be accessed. Validate the stream is not disposed before it's used. | 20 | ok |
|  | Function | Check that instances of classes are initialized in other classes that use the information | 21 | ok |
|  | Function | Check that result statements are not inside loops or conditional statements in order to avoid updating and calculating results for each set of data when it is not necessary | 20 | ok |
|  | Function | Do not use reflection | 27, 32 | ok |
|  | Interface | Validate usefulness and correctness of user messages |  | ok |
|  | Interface | Make sure program does not exit on invalid input without warning for the user | 9,19, 31 | ok |
| 35 | Function | Validate correct and complete use of indexing in arrays and lists | 1, 30 | ok |
|  | Function | Validate string operations. User String.Empty instead of space. Use Trim() function to avoid misreading | 13 | ok |
| 40 | Interface | Make sure to use correct types – int versus double |  | ok |
| 42, 46 | Interface | Do not pass parameters to functions when they have to be modified and then used by the function, otherwise old values are taken into account |  | ok |
| 43, | Assignment | Do not initialize objects in the constructor of the classes because you don't understand how it's used yet |  | ok |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |