#### Design Review Checklist Template

|  |  |  |  |
| --- | --- | --- | --- |
| Student | Hector Manuel Takami Flores | Date | 07/04/18 |
| Program | 7 | Program # | 7 |
| Instructor | Adriana Bojorquez | Language | JAVA |

|  |  |
| --- | --- |
| Purpose | To guide you in conducting an effective design 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. |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | I/O | A | B0 | B1 | R | M | G | f | TA | C | X | S | R |
| COMPLETE | * All specified outputs are produced. * All needed inputs are furnished. * All required includes are stated. | Ok | Ok | Ok | ok | Ok | Ok | ok | ok | ok | Ok | Ok | Ok | Ok |
| EXTERNAL LIMITS | Where the design assumes or relies upon external limits, determine if behavior is correct at nominal values, at limits, and beyond limits. | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok |
| LOGIC | * Verify that all loops are properly initiated, incremented, and terminated. * Examine each conditional statement and verify all cases. * Verify that types are well used | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok |
| INTERNAL LIMITS | Where the design assumes or relies upon internal limits, determine if behavior is correct at nominal values, at limits, and beyond limits. | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok |
| SPECIAL CASES | * Ensure proper operation with empty, full, minimum, maximum, negative, and zero values for all variables. * Protect against out-of-limits, overflow, and underflow conditions. * Ensure “impossible” conditions are absolutely impossible.   Handle all possible incorrect or error conditions. | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok |
| FUNCTIONAL USE | - Verify that all functions, procedures, or methods are fully understood and properly used.  - Verify that all externally referenced abstractions are precisely defined.  -Verify that there aren´t unused functions, procedures, or methods. | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok |
| SYSTEM CONSIDERATIONS | -Verify that the program does not cause system limits to be exceeded.  -Verify that all security-sensitive data are from trusted sources.  -Verify that all safety conditions conform to the safety specifications. | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok |
| NAMES | Verify that   * Variable names are used properly, and with logic * the scopes of all variables and parameters are self-evident or defined   -all named items are used within their declared scopes | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok |
| STANDARDS | Ensure that the design conforms to all applicable design standards. | ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok | Ok |