# Code Review Checklist:

*Prepared by Team Ocelots*

**Grading** – In each category, you have 1-5 grading and you can use it to grade the team based on the category. 1 means you did not meet the grading standard of the category. 3 means you have all the elements of what the category grades on. 5 means you meet and exceeded the grading standard of the category.

**Evaluation Criteria**

1. **Code Design**

* How does the code fit with the overall architecture?

2. **Proper Commenting**

* Are confusing sections of code either documented, commented, or covered by understandable tests?

3. **Coding Style**

* How is the formatting, style and naming conventions?

4. **Coding Conventions**

* Does the code match their code conventions they listed?

5. **Areas of concern**

* Are there potential security problems with the code? Are there requirements that need to be met or anything you can think about?

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Team | Evaluation Criteria | | | | | |
|  | Coding Design | Proper Commenting | Coding Style | Coding Convention | Area of Concern | Comment  By team Ocelots |
| Wildcats | 5 | 4 | 5 | 5 | 5 | Overall, you have a very nice code well done. You follow the coding convention which is formatting, style and naming convention which are camel case when defining method and the variable name Using the private modifier is protects the code not to be access so did a great job. You have a java doc with necessary information and some comments on the code but if you add some inline comment helps other to understand better about the code. Breaking a method in to a smaller method will make the to be easily manageable so it better to break your method instead of using big method. |
| Trinity of Awesomeness | 5 | 3 | 5 | 5 | 5 | Great work, your code is simple and well organized. You follow the coding convention for class name, variable name, the parameter, and method signature. You have getter and setter method and that make your code to hide the internal representation of the property while exposing a property using an alternative representation. Having a java doc and commenting will help for future reference so adding these would be good. |
| Quasar | 5 | 5 | 5 | 5 | 5 | Great work, it is well organized and understandable code. You put a java documentation for the classes and the method and uses the proper naming convention throughout your code and that makes easy for the one who uses this code in the future. In each class you have a few numbers of lines of code which shows us how efficient your code is. You have getter and setter method and that make your code to hide the internal representation of the property while exposing a property using an alternative representation. |
| Bobcats | 5 | 5 | 5 | 5 | 5 | Great work, it is well organized, simple, and understandable code. You put a java documentation and comment for the class and methods. Nice use formatting and proper naming convention throughout your code and that makes easy for the one who uses this code in the future. Appropriate use of access modifier. |
| Tiger | 5 | 5 | 5 | 5 | 5 | Great work, it is well organized, simple, and understandable code. You put a java documentation and comments for the class and methods. Nice use formatting and proper naming convention throughout your code and that makes easy for the one who uses this code in the future. Using private modifier not to access information about the user is good and appropriate well done. |
| Dragon | 5 | 5 | 5 | 5 | 1 | Great work, you have a java documentation with author tag and with explanation of the use of the class and the method. You follow the naming convention of java code. Your coding format is good and that make the code readable. The class seem a little bet long and encapsulate your class by using setter and getter method or using a private access modifier will help your class not to be accessed. |
| SASS | 5 | 4 | 5 | 5 | 5 | Great work, it is well organized, simple, and understandable code. Nice use formatting and proper naming convention throughout your code and that makes easy for the one who uses this code in the future. Appropriate use of access modifier. Adding some a java documentation and comment for all the class and methods helps for the future reference. |