# **Project Sheet**

|  |  |
| --- | --- |
| **Course and instructor name** | CPAN 131 – Dina Goldenberg |
| **Assignment name** | Project |
| **Grade value** | 10%  Rubric attached |
| **Due date** | Week 13 |
| **Individual or group assignment** | Group (no individual submissions) |
| **Targeting these learning outcomes from course outline** | Practicing OOP concepts in Java |
| **Instructions** | Prepare pieces for chess game. All the pieces should have the following two methods:   * **boolean canMoveTo(int x, int y)** which will be checking if this piece can move to the given coordinates * **moveTo(int x, int y)** that can move the piece only if it is legal     Place all the pieces on the board and start a two-player game. Each player enters two sets of coordinates of the piece they want to move – origin and destination. The game must check if there is a piece present and if it can move to the requested coordinates. If successful, the piece is moved, and the updated board is presented.  (No need to implement checkmates and checks)  Proper use of OOP principles is required – encapsulation, inheritance and polymorphism. |
| **Submission** | Through Backboard |
| **Submission Details** | Submit a zip file of all of the NetBeans directory of your project.  Incomplete projects, different IDE projects or separate code files will not be evaluated.  Indicate all the members of the group. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Rubric – Project | | | | | |
|  | 20% | 15% | 10% | 5% | 0% |
| **OOP principles** | Proper inheritance hierarchy with implemented concepts of polymorphism are present | Proper inheritance hierarchy is presented | Some inheritance is implemented | Encapsulation is present | Missing |
| **Pieces classes** | Coded extremely well and wonderful comments | Coded to specifications | Incorrect functionality | Missing most details | Largely empty |
| **Execution** | Application is fully input proof | Most of foreseen cases are taken care of | Substantial number of Runtime errors | Execution crashes pretty quickly | Project doesn’t compile |
| **Output** | Console output is well formatted to see clearly code working as expected | Coded to specifications | Missing or broken functionality | Missing lots of detail | Largely empty |
| **Process** | Pieces of the assignment have been steadily building throughout the time given. Creativity used to solve key problems. Able to reflect on what they could improve on. | Assignment completed satisfactorily | Somewhat of a rush to complete the assignment | Obviously has been a big last-minute rush to complete assignment. No time for creativity or reflection | Students have not spent any time grappling with concepts, reaching out for help, or reflecting on topic |