**SE-class project To-do list**

1. Understand the overall project.
2. Read and understand the research paper provided.
3. Understand the dependencies of the project file.
4. Plan for necessary setup.
5. Planning to choose the right environment.
6. Change the product backlog after finding the dependencies.
7. Understand the working of all Java files.
8. Understand the working of compiler dependencies files.
9. Compile all the java programs for the prolog Machine.
10. Execute add.pl,big.pl,lambdas.pl,mperms.pl,perms.pl,queens.pl,sud4x.pl and check the output which are useful later for comparison.
11. Understand the implementation of given java program Clause.java
12. Understand the implementation of given java program Engine.java
13. Understand the implementation of given java program Imap.java
14. Understand the implementation of given java program Intlist.java
15. Understand the implementation of given java program IntMap.java
16. Understand the implementation of given java program IntStack.java
17. Understand the implementation of given java program Main.java
18. Understand the implementation of given java program Obstack.java
19. Understand the implementation of given java program Prog.java
20. Understand the implementation of given java program Spine.java
21. Understand the implementation of given java program Toks.java
22. Implement the java program Clause.java to either C or C++ program depending on the language chosen.
23. Implement the java program Engine.java to either C or C++ program depending on the language chosen.
24. Implement the java program Imap.java to either C or C++ program depending on the language chosen.
25. Implement the java program Intlist.java to either C or C++ program depending on the language chosen.
26. Implement the java program IntMap.java to either C or C++ program depending on the language chosen.
27. Implement the java program IntStack.java to either C or C++ program depending on the language chosen.
28. Implement the java program Main.java to either C or C++ program depending on the language chosen.
29. Implement the java program Obstack.java to either C or C++ program depending on the language chosen.
30. Implement the java program Prog.java to either C or C++ program depending on the language chosen.
31. Implement the java program Spine.java to either C or C++ program depending on the language chosen.
32. Implement the java program Toks.java to either C or C++ program depending on the language chosen.
33. Perform testing for the Clause.CPP file
34. Perform testing for the Engine.CPP file
35. Perform testing for the Imap.CPP file
36. Perform testing for the Intlist.CPP file
37. Perform testing for the IntMap.CPP file
38. Perform testing for the IntStack.CPP file
39. Perform testing for the Main.CPP file
40. Perform testing for the Obstack.CPP file
41. Perform testing for the prog.CPP file
42. Perform testing for the Spine.CPP file
43. Perform testing for the Toks.CPP file
44. Test the C++ compiler by combining all the C++ code.
45. Execute add.pl prolog program with C++ compiler and compare it with the java compiler output to verify the correctness.
46. Execute big.pl prolog program with C++ compiler and compare it with the java compiler output to verify the correctness.
47. Execute lambdas.pl prolog program with C++ compiler and compare it with the java compiler output to verify the correctness.
48. Execute mperms.pl prolog program with C++ compiler and compare it with the java compiler output to verify the correctness.
49. Execute perms.pl prolog program with C++ compiler and compare it with the java compiler output to verify the correctness.
50. Execute queens.pl prolog program with C++ compiler and compare it with the java compiler output to verify the correctness.
51. Execute sud4X.pl prolog program with C++ compiler and compare it with the java compiler output to verify the correctness.