



# Informatics Institute of Technology

Department Of Computing

**Module: ECSI410 – Software Development Principles 01**

**Module Leader: Mr. Guganathan Poravi**

**Coursework**

Date of submission: 27/04/2016

Student ID: w1608490

Student First Name: Chamod Oshan

Student Surname: Hettigodage

## Content

• Acknowledgement	02
• Introduction	02
• List of Functional Requirements	02
• Non-Functional Requirements	02
• Add a student	03
• Create the level 04/05/06 file and write marks	03
• Read the level 04/05/06 file , calculate credits and save the result to the Level4/5/6 Status file	04
• Read the Level4/5/6 Status file	05
• Exit menu for Cer HE/Dip HE	05
• Calculate number of resits and retakes	06
• Calculate Award and write to the Award Status file	06
• Generate the Report	07
• Add next student	08
• Conclusion	15

## Figures

• Figure 1.0 - Screenshot of For Frist Test cases	10
• Figure 1.1 - Screenshot of Final Award Calculation	11-13
• Figure 1.2 - Screenshot of Final Report	14

## Table

• Table 1.0 - White Box Testing 1	9
• Table 1.1 - White Box Testing 2	9
• Table 1.2 - Black Box Testing	14-15

# Acknowledgment

My sincere gratitude goes to our module leader Mr.Guhanathan Poravi for providing us with the knowledge and guiding us as we were accomplish this task. My parents for their endless support and encouragement without whom I may not be where I am today.

## Introduction

This CLI based java program is created as a Course work of ECSI410 – Software Development Principles 01. This program build to do Awards calculation of a university using student marks. This program designed to get exam marks for each module. After it calculate average of each module and show whether the module is Pass ,Resit or Retake. Then calculate credits in the Leve04,If credits enough to move to the next level, program will go to next level or if you want exit with CerHE /DipHE you can exit with enough credits . At last this program will show the degree , class of degree, by concerning about the total credits .Then if you want to add a student you can add students continuously .Finally this program will generate a text report by students details.

### List of Functional Requirements

- Login to the program.
- Calculate the Level04 credits and marks.
- Exit with CerHE.
- Calculate the Level05 credits and marks.
- Exit with DipHE.
- Calculate Level06 credits and marks.
- Final Award calculation.

### Non-Functional Requirements

- Design of gui.
- Number of buttons.
- Font size and font face.
- Switching animations.
- Number of windows.

## ❖ Add a student

```
58      System.out.println("");
59      Scanner sc=new Scanner(System.in);
60      File stu = new File ("Student.txt");
61      System.out.println("Enter Student Name and Id:-");
62      String name = sc.next();
63      int id =sc.nextInt();
64      Student S1 =new Student(name,id);
65      try {
66          FileOutputStream fos = new FileOutputStream(stu);
67          ObjectOutputStream oos = new ObjectOutputStream(fos);
68          oos.writeObject(S1);
69          oos.flush();
70          oos.close();
71          fos.close();
72      } catch (IOException e) {
73          e.printStackTrace();
74      }
```

## ❖ Create the level 04/05/06 file and write marks

```
75      System.out.println("");
76      System.out.println("---Level 04---");
77      System.out.println("");
78      File file = new File ("Level04.txt");
79
80      //module 01
81      System.out.println("Module 1:-");
82      int [] M1ict=new int[3];
83      for (int i =0; i<3;i++){
84          System.out.print("ICT"+(i+1)+" : ");
85          M1ict [i]=sc.nextInt();
86          M1I1=M1ict[0];
87          M1I2=M1ict[1];
88          M1I3=M1ict[2];
89      }
90      total=M1I1+M1I2+M1I3;
91      avg1=total/3;
92      Marks M1 =new Marks(avg1);

```

```
165      try {
166          FileOutputStream fos = new FileOutputStream(file);
167          ObjectOutputStream oos = new ObjectOutputStream(fos);
168          oos.writeObject(M1);
169          oos.writeObject(M2);
170          oos.writeObject(M3);
171          oos.writeObject(M4);
172          oos.writeObject(M5);
173          oos.writeObject(M6);
174          oos.flush();
175          oos.close();
176          fos.close();
177      } catch (IOException e) {
178          e.printStackTrace();
179      }
---
```

❖ Read the level 04/05/06 file , calculate credits and save the result to the  
Level4/5/6 Status file

```
182      FileInputStream fis;
183      try {
184          fis = new FileInputStream(file);
185          ObjectInputStream ois = new ObjectInputStream(fis);
186          try {
187              Marks T1 = (Marks)ois.readObject();
188              Marks T2 = (Marks)ois.readObject();
189              Marks T3 = (Marks)ois.readObject();
190              Marks T4 = (Marks)ois.readObject();
191              Marks T5 = (Marks)ois.readObject();
192              Marks T6 = (Marks)ois.readObject();
193
194              File level04 = new File("Level4 Status.txt");
195              PrintWriter pw=null;
196              FileWriter fw=null;
197              try {
198                  fw=new FileWriter(level04);
199                  pw=new PrintWriter(fw,true); // true Stands for auto flush
200
201
202
203              //Module 01
204              System.out.println("Module 1:-");
205              System.out.println(T1);
206
207              if (avg1 >= 40&&avg1<=100 && (MI11 < 40 || MI12 < 40 || MI13 < 40)) {
208                  System.out.println("PASS");
209                  System.out.println("CONDONED");
210                  pw.println("Module 01:-"+avg1+" PASS-CONDONED");
211                  con=con+20;
212              } else if (avg1 >= 40&&avg1<=100) {
213                  System.out.println("PASS");
214                  pw.println("Module 01:-"+avg1+" PASS");
215                  credit1=credit1+20;
216              } else if (avg1 >= 30&&avg1<=100) {
217                  System.out.println("RESIT");
218                  System.out.println("CONDONED");
219                  pw.println("Module 01:-"+avg1+" RESIT-CONDONED");
220                  con=con+20;
221              } else if (avg1 < 30) {
222                  System.out.println("RETAKE");
223                  pw.println("Module 01:-"+avg1+" RETAKE");
224                  System.out.println("");
225                  count=count+1;
226
227              }else{
228                  System.out.println("Invalid");
229              }
230          }
231      }
232      catch (IOException e) {
233          e.printStackTrace();
234      }
235      finally {
236          if (ois != null) {
237              ois.close();
238          }
239          if (fis != null) {
240              fis.close();
241          }
242      }
243      }
244      }
245      }
246      }
247      }
248      }
249      }
250      }
251      }
252      }
253      }
254      }
255      }
256      }
257      }
258      }
259      }
260      }
261      }
262      }
263      }
264      }
265      }
266      }
267      }
268      }
269      }
270      }
271      }
272      }
273      }
274      }
275      }
276      }
277      }
278      }
279      }
280      }
281      }
282      }
283      }
284      }
285      }
286      }
287      }
288      }
289      }
290      }
291      }
292      }
293      }
294      }
295      }
296      }
297      }
298      }
299      }
300      }
301      }
302      }
303      }
304      }
305      }
306      }
307      }
308      }
309      }
310      }
311      }
312      }
313      }
314      }
315      }
316      }
317      }
318      }
319      }
320      }
321      }
322      }
323      }
324      }
325      }
326      }
327      }
328      }
329      }
330      }
331      }
332      }
333      }
334      }
335      }
336      }
337      }
338      }
339      }
340      }
341      }
342      }
343      }
344      }
345      }
346      }
347      }
348      }
349      }
350      }
351      }
352      }
353      }
354      }
355      }
356      }
357      }
358      }
359      }
360      }
361      }
362      }
363      }
364      }
365      }
366      }
367      }
368      }
369      }
370      }
371      }
372      }
373      }
374      }
375      }
376      }
377      }
378      }
379      }
380      }
381      }
382      }
383      }
384      }
385      }
386      }
387      }
388      }
389      }
390      }
391      }
392      }
393      }
394      }
395      }
396      }
397      }
398      }
399      }
400      }
401      }
402      }
403      }
404      }
405      }
406      }
407      }
408      }
409      }
410      }
411      }
412      }
413      }
414      }
415      }
416      }
417      }
418      }
419      }
420      }
421      }
422      }
423      }
424      }
425      }
426      }
427      }
428      }
429      }
430      }
431      }
432      }
433      }
434      }
435      }
436      }
437      }
438      }
439      }
440      }
441      }
442      }
443      }
444      }
445      }
446      }
447      }
448      }
449      }
450      }
451      }
452      }
453      }
454      }
455      }
456      }
457      }
458      }
459      }
460      }
461      }
462      }
463      }
464      }
465      }
466      }
467      }
468      }
469      }
470      }
471      }
472      }
473      }
474      }
475      }
476      }
477      }
478      }
479      }
480      }
481      }
482      }
483      }
484      }
485      }
486      }
487      }
488      }
489      }
490      }
491      }
492      }
493      }
494      }
495      }
496      }
497      }
498      }
499      }
500      }
501      }
502      }
503      }
504      }
505      }
506      }
507      }
508      }
509      }
510      }
511      }
512      }
513      }
514      }
515      }
516      }
517      }
518      }
519      }
520      }
521      }
522      }
523      }
524      }
525      }
526      }
527      }
528      }
529      }
530      }
531      }
532      }
533      }
534      }
535      }
536      }
537      }
538      }
539      }
540      }
541      }
542      }
543      }
544      }
545      }
546      }
547      }
548      }
549      }
550      }
551      }
552      }
553      }
554      }
555      }
556      }
557      }
558      }
559      }
560      }
561      }
562      }
563      }
564      }
565      }
566      }
567      }
568      }
569      }
570      }
571      }
572      }
573      }
574      }
575      }
576      }
577      }
578      }
579      }
580      }
581      }
582      }
583      }
584      }
585      }
586      }
587      }
588      }
589      }
590      }
591      }
592      }
593      }
594      }
595      }
596      }
597      }
598      }
599      }
600      }
601      }
602      }
603      }
604      }
605      }
606      }
607      }
608      }
609      }
610      }
611      }
612      }
613      }
614      }
615      }
616      }
617      }
618      }
619      }
620      }
621      }
622      }
623      }
624      }
625      }
626      }
627      }
628      }
629      }
630      }
631      }
632      }
633      }
634      }
635      }
636      }
637      }
638      }
639      }
640      }
641      }
642      }
643      }
644      }
645      }
646      }
647      }
648      }
649      }
650      }
651      }
652      }
653      }
654      }
655      }
656      }
657      }
658      }
659      }
660      }
661      }
662      }
663      }
664      }
665      }
666      }
667      }
668      }
669      }
670      }
671      }
672      }
673      }
674      }
675      }
676      }
677      }
678      }
679      }
680      }
681      }
682      }
683      }
684      }
685      }
686      }
687      }
688      }
689      }
690      }
691      }
692      }
693      }
694      }
695      }
696      }
697      }
698      }
699      }
700      }
701      }
702      }
703      }
704      }
705      }
706      }
707      }
708      }
709      }
710      }
711      }
712      }
713      }
714      }
715      }
716      }
717      }
718      }
719      }
720      }
721      }
722      }
723      }
724      }
725      }
726      }
727      }
728      }
729      }
730      }
731      }
732      }
733      }
734      }
735      }
736      }
737      }
738      }
739      }
740      }
741      }
742      }
743      }
744      }
745      }
746      }
747      }
748      }
749      }
750      }
751      }
752      }
753      }
754      }
755      }
756      }
757      }
758      }
759      }
760      }
761      }
762      }
763      }
764      }
765      }
766      }
767      }
768      }
769      }
770      }
771      }
772      }
773      }
774      }
775      }
776      }
777      }
778      }
779      }
780      }
781      }
782      }
783      }
784      }
785      }
786      }
787      }
788      }
789      }
790      }
791      }
792      }
793      }
794      }
795      }
796      }
797      }
798      }
799      }
800      }
801      }
802      }
803      }
804      }
805      }
806      }
807      }
808      }
809      }
810      }
811      }
812      }
813      }
814      }
815      }
816      }
817      }
818      }
819      }
820      }
821      }
822      }
823      }
824      }
825      }
826      }
827      }
828      }
829      }
830      }
831      }
832      }
833      }
834      }
835      }
836      }
837      }
838      }
839      }
840      }
841      }
842      }
843      }
844      }
845      }
846      }
847      }
848      }
849      }
850      }
851      }
852      }
853      }
854      }
855      }
856      }
857      }
858      }
859      }
860      }
861      }
862      }
863      }
864      }
865      }
866      }
867      }
868      }
869      }
870      }
871      }
872      }
873      }
874      }
875      }
876      }
877      }
878      }
879      }
880      }
881      }
882      }
883      }
884      }
885      }
886      }
887      }
888      }
889      }
890      }
891      }
892      }
893      }
894      }
895      }
896      }
897      }
898      }
899      }
900      }
901      }
902      }
903      }
904      }
905      }
906      }
907      }
908      }
909      }
910      }
911      }
912      }
913      }
914      }
915      }
916      }
917      }
918      }
919      }
920      }
921      }
922      }
923      }
924      }
925      }
926      }
927      }
928      }
929      }
930      }
931      }
932      }
933      }
934      }
935      }
936      }
937      }
938      }
939      }
940      }
941      }
942      }
943      }
944      }
945      }
946      }
947      }
948      }
949      }
950      }
951      }
952      }
953      }
954      }
955      }
956      }
957      }
958      }
959      }
960      }
961      }
962      }
963      }
964      }
965      }
966      }
967      }
968      }
969      }
970      }
971      }
972      }
973      }
974      }
975      }
976      }
977      }
978      }
979      }
980      }
981      }
982      }
983      }
984      }
985      }
986      }
987      }
988      }
989      }
990      }
991      }
992      }
993      }
994      }
995      }
996      }
997      }
998      }
999      }
1000     }
```

## ❖ Read the Level4/5/6 Status file

```
397         try {
398             Scanner sc2 = new Scanner(level04);
399             l4m1Rep = sc2.next();
400             l4m2Rep = sc2.next();
401             l4m3Rep = sc2.next();
402             l4m4Rep = sc2.next();
403             l4m5Rep = sc2.next();
404             l4m6Rep = sc2.next();
405             l4m7Rep = sc2.next();
406             l4m8Rep = sc2.next();
407             l4m9Rep = sc2.next();
408             l4m10Rep = sc2.next();
409             l4m11Rep = sc2.next();
410             l4m12Rep = sc2.next();
411             l4m13Rep = sc2.next();
412             l4m14Rep = sc2.next();
413             l4m15Rep = sc2.next();
414             l4m16Rep = sc2.next();
415             l4m17Rep = sc2.next();
416             l4m18Rep = sc2.next();
417
418         } catch (FileNotFoundException e) {
419             e.printStackTrace();
420         }
```

## ❖ Exit menu for Cer HE/Dip HE

```
428         System.out.println("");
429         System.out.println("Choose Your Career:-");
430         System.out.println("(1)Continue");
431         System.out.println("(2)Exit with CerHE");
432         int cerhe=sc.nextInt();
433         if(cerhe==1) {

842     }else{
843         System.out.println("-----");
844         System.out.println("|Total Credits = "+credit1+" |");
845         System.out.println("|YOU HAVE THE CERTIFICATE OF HIGHER EDUCATION |");
846         System.out.println("-----");
847     }
```

## ❖ Calculate number of resits and retakes

```
589 //Module 01
590 System.out.println("Module 1:-");
591 System.out.println(T1);
592
593 if (L5avg1 >= 40&&L5avg1<=100 && (M1I1 < 40 || M1I2 < 40 || M1I3 < 40)) {
594     System.out.println("RESIT");
595     pw.println("Module 01:-"+L5avg1+" RESIT");
596     count2=count2+1;
597 } else if (L5avg1 >= 40&&L5avg1<=100) {
598     System.out.println("PASS");
599     pw.println("Module 01:-"+L5avg1+" PASS");
600     credit2=credit2+20;
601 } else if (L5avg1 >= 30&&L5avg1<=100) {
602     System.out.println("RESIT");
603     pw.println("Module 01:-"+L5avg1+" RESIT");
604     count2=count2+1;
605 } else if (L5avg1 < 30) {
606     System.out.println("RETAKE");
607     System.out.println("");
608     pw.println("Module 01:-"+L5avg1+" RETAKE");
609     count=count+1;
610
611 }else{
612     System.out.println("Invalid");
613 }
```

## ❖ Calculate Award and write to the Award Status file

```
1219 if(credit3==120&&L5avgR<40||credit3==120||credit3 >= 60 && L5avgR < 40||credit3 == 140){
1220     int totalCredit=credit1+credit2+credit3;
1221
1222     File award = new File("Award Status.txt");
1223     PrintWriter pw=null;
1224     FileWriter fw=null;
1225     try {
1226         fw=new FileWriter(award);
1227         pw=new PrintWriter(fw,true); // true Stands for auto flush
1228
1229         min=100;
1230         List <Integer> avg =new ArrayList <Integer> ();
1231         avg.add(L5avg1);
1232         avg.add(L5avg2);
1233         avg.add(L5avg3);
1234         avg.add(L5avg4);
1235         avg.add(L5avg5);
1236         avg.add(L5avg6);
1237         avg.add(L6avg1);
1238         avg.add(L6avg2);
1239         avg.add(L6avg3);
1240         avg.add(L6avg4);
1241         avg.add(L6avg5);
1242         avg.add(L5avgR);
1243         total=0;
1244         for (int value:avg) {
1245             if(value>=40){
1246                 total=total+value;
1247                 if (value <= min) {
1248                     min = value;
1249                 }
1250             }
1251         }
1252     }
```

```

1253     total=total-min;
1254     int totalAvg=0;
1255     totalAvg=total/10;
1256     System.out.println("");
1257     System.out.println("-----");
1258     System.out.println("    |Total Credits = "+totalCredit+"    |");
1259     System.out.println("    |Total Average = "+totalAvg+"    |");
1260     pw.println("Total Credits = "+totalCredit);
1261     pw.println("Total Average = "+totalAvg);
1262
1263
1264     if (totalCredit == 360) {
1265         if (totalAvg >= 70) {
1266             System.out.println("    |Class:-1st Class Honors    |");
1267             System.out.println("    |Degree:-B.Sc. 1st Class Honors DEGREE    |");
1268             pw.println("Class:-1st Class Honors");
1269             pw.println("Degree:-B.Sc. 1st Class Honors DEGREE");
1270         } else if (totalAvg >= 60) {
1271             System.out.println("    |Class:-2nd Class Honors Upper Division    |");
1272             System.out.println("    |Degree:-B.Sc. 2nd Class Honors Upper Division DEGREE|");
1273             pw.println("Class:-2nd Class Honors Upper Division");
1274             pw.println("Degree:-B.Sc. 2nd Class Honors Upper Division DEGREE");
1275         } else if (totalAvg >= 50) {
1276             System.out.println("    |Class:-2nd Class Honors Lower Division    |");
1277             System.out.println("    |Degree:-B.Sc. 2nd Class Honors Lower Division DEGREE|");
1278             pw.println("Class:-2nd Class Honors Lower Division");
1279             pw.println("Degree:-B.Sc. 2nd Class Honors Lower Division DEGREE");
1280         } else if (totalAvg >= 40) {
1281             System.out.println("    |Class:-3rd Class Honors    |");
1282             System.out.println("    |Degree:-B.Sc. 3rd Class Honors DEGREE    |");
1283             pw.println("Class:-3rd Class Honors");
1284             pw.println("Degree:-B.Sc. 3rd Class Honors DEGREE");
1285         } else {
1286             System.out.println("    |Class:-No Class    |");
1287             System.out.println("    |Degree:-B.Sc. Honors DEGREE    |");
1288             pw.println("Class:-No Class");
1289             pw.println("Degree:-B.Sc. Honors DEGREE");
1290         }
1291     } else if (totalCredit >= 300) {
1292         System.out.println("    |Degree:-B.Sc. DEGREE    |");
1293         pw.println("Degree:-B.Sc. DEGREE");
1294     }
1295
1296     System.out.println("-----");
1297
1298
1299
1300     } catch (IOException e) {
1301
1302         e.printStackTrace();
1303     } finally{
1304         pw.close();
1305     }

```

## ❖ Generate the Report

```

1342     FileInputStream fiss;
1343     try {
1344         fiss = new FileInputStream(stu);
1345         ObjectInputStream ois = new ObjectInputStream(fiss);
1346         try {
1347             Student Add1 = (Student)ois.readObject();
1348
1349             File report = new File("Report.txt");
1350             PrintWriter pw=null;
1351             FileWriter fw=null;
1352             try {
1353                 fw=new FileWriter(report,true);//TO appending without override
1354                 pw=new PrintWriter(fw,true); // true Stands for auto flush
1355                 pw.println(Add1);
1356                 pw.println("");
1357                 pw.print("Level-04    ");
1358                 pw.print("Level-05    ");
1359                 pw.println("Level-06    ");
1360                 pw.print("14m1Rep+" );
1361                 pw.print("14m2Rep+" );
1362                 pw.print("14m3Rep+" );
1363                 pw.print("15m1Rep+" );
1364                 pw.print("15m2Rep+" );
1365                 pw.print("15m3Rep+" );
1366                 pw.print("16m1Rep+" );
1367                 pw.print("16m2Rep+" );
1368                 pw.println("16m3Rep");
1369                 pw.print("14m4Rep+" );
1370                 pw.print("14m5Rep+" );
1371                 pw.print("14m6Rep+" );
1372                 pw.print("15m4Rep+" );
1373                 pw.print("15m5Rep+" );
1374                 pw.print("15m6Rep+" );
1375                 pw.print("16m4Rep+" );
1376                 pw.print("16m5Rep+" );
1377                 pw.println("16m6Rep");
1378                 pw.print("14m7Rep+" );
1379                 pw.print("14m8Rep+" );
1380                 pw.print("14m9Rep+" );

```



```

1381         pw.print(15m7Rep+" ");
1382         pw.print(15m8Rep+" ");
1383         pw.print(15m9Rep+" ");
1384         pw.print(16m7Rep+" ");
1385         pw.print(16m8Rep+" ");
1386         pw.println(16m9Rep);
1387         pw.print(14m10Rep+" ");
1388         pw.print(14m11Rep+" ");
1389         pw.print(14m12Rep+" ");
1390         pw.print(15m10Rep+" ");
1391         pw.print(15m11Rep+" ");
1392         pw.print(15m12Rep+" ");
1393         pw.print(16m10Rep+" ");
1394         pw.print(16m11Rep+" ");
1395         pw.println(16m12Rep);
1396         pw.print(14m13Rep+" ");
1397         pw.print(14m14Rep+" ");
1398         pw.print(14m15Rep+" ");
1399         pw.print(15m13Rep+" ");
1400         pw.print(15m14Rep+" ");
1401         pw.print(15m15Rep+" ");
1402         pw.print(16m13Rep+" ");
1403         pw.print(16m14Rep+" ");
1404         pw.println(16m15Rep);
1405         pw.print(14m16Rep+" ");
1406         pw.print(14m17Rep+" ");
1407         pw.print(14m18Rep+" ");
1408         pw.print(15m16Rep+" ");
1409         pw.print(15m17Rep+" ");
1410         pw.println(15m18Rep+" ");
1411         pw.print(15m19Rep+" ");
1412         pw.print(15m20Rep+" ");
1413         pw.print(15m21Rep+" ");
1414         pw.print(16m16Rep+" ");
1415         pw.print(16m17Rep+" ");
1416         pw.println(16m18Rep);
1417
1418         if(credit3>=60){
1419             pw.println(award1);
1420             pw.println(award2);
1421             pw.println(award3);
1422             pw.println(award4);
1423             pw.println(award5);
1424             pw.println(award6);
1425             pw.println(award7);
1426             pw.println(award8);
1427             pw.println(award9);
1428             pw.println(award10);
1429             pw.println(award11);
1430             pw.println(award12);
1431             pw.println(award13);
1432             pw.println(award14);
1433             pw.println(award15);
1434             pw.println(award16);
1435         }
1436         if(count+count2>2){
1437             pw.println("RESIT Modules = "+count2);
1438             pw.println("RETAKE Modules = "+count);
1439         }
1440     }
1441     //write report details***
1442
1443 } catch (IOException e) {
1444     e.printStackTrace();
1445 }finally{
1446     pw.close();
1447 }
1448
1449

```

## ❖ Add next student

```

1463     int i=1;
1464     System.out.println("");
1465     System.out.println("Choose Your desire:-");
1466     System.out.println("(1)Add a Student");
1467     System.out.println("(2)Finish");
1468
1469     num=sc.nextInt();
1470     while(num==1);
1471

```

## Test Cases

### Black-Box Testing

INPUT	EXPECTED RESULT	ACTUAL RESULT
<b>Enter Student Name &amp; Id</b>	It will be able to enter only One student .	It will be able to enter only One student .
<b>Enter Marks For Each Module</b>	It will be able to enter only Integers.	It will be able to enter only Integers.
<b>Enter 1 When Ask To Continue</b>	Move to the next level	Move to the next level
<b>Enter 2 When Ask To Continue</b>	Exit with Cer HE/Dip HE	Exit with Cer HE/Dip HE
<b>Enter 1 When Ask To Continue After Award Calculation</b>	Move to Add Student statement .	Move to Add Student statement .
<b>Enter 2 When Ask To Continue After Award Calculation</b>	End the program .	End the program .

Table 1.0

INPUT	EXPECTED RESULT	ACTUAL RESULT
<b>45 855 799</b>	invalid	invalid
<b>45 85 79</b>	Average = 69% , Pass	Average = 69% , Pass
<b>80 80 20</b>	Average = 60% ,L4- Pass, con L5/L6-Resit	Average = 60% ,L4- Pass, con L5/L6-Resit
<b>10 10 10</b>	Average = 10% ,Retake	Average = 10% ,Retake
<b>40 40 40</b>	Average = 40% , Pass	Average = 40% , Pass
<b>39 39 39</b>	Average = 39% , L4-Resit, con L5/L6-Resit	Average = 39% , L4-Resit, con L5/L6-Resit
<b>100 100 100</b>	Average = 100% , Pass	Average = 100% , Pass
<b>Retake L04:-85 99 78</b>	Average = 40% , L5/L6-Pass	Average = 40% , L5/L6-Pass
<b>Final Award Calculation</b>	Credits=360, Average=79, Degree=B.Sc.Honors, Degree Class= 1 <sup>st</sup> Class	Credits=360, Average=79, Degree=B.Sc.Honors, Degree Class= 1 <sup>st</sup> Class

Table 1.1

## For Frist Test cases

```

**  **  ***  **  *****  **      **  *****  *****  *****  *****  *****  **  **
**  **  **  *  **  **      **      **  **      **  **      **      **      **      **  **
**  **  **  *  **  **      **      **  **      *****  **  **      *****  **      **      ****
**  **  **  *  **  **      **      **  **      **      **  **      **      **      **      **
*****  **  ***  *****      ***      *****  **  **      *****  *****  **      **
-----
*****  *****  *****  **  **  *****  *****  *****
**  **  **      **      **      **      **      **
**  **  *****  **  *****  **      **  *****  *****  **
**  **  **      **  **  **  **      **  **  **      **  **
*****  **      *****      *****  *****  *****  *****
-----

```

Enter Student Name and Id:-

chamod  
2015300

---Level 04---

Module 1:-

ICT1 : 45  
ICT2 : 855  
ICT3 : 799

Module 2:-

ICT1 : 45  
ICT2 : 85  
ICT3 : 79

Module 3:-

ICT1 : 10  
ICT2 : 10  
ICT3 : 10

Module 4:-

ICT1 : 40  
ICT2 : 40  
ICT3 : 40

Module 5:-

ICT1 : 39  
ICT2 : 39  
ICT3 : 39

Module 6:-

ICT1 : 100  
ICT2 : 100  
ICT3 : 100

Module 1:-

Average = 566

Invalid

Module 2:-

Average = 69

PASS

Module 3:-

Average = 10

RETAKE

Module 4:-

Average = 40

PASS

Module 5:-

Average = 39

RESIT

CONDONED

Module 6:-

Average = 100

PASS

Credits = 80

You are not eligible for move to the Level-05

Exception in thread "main" java.util.NoSuchElementException

at java.util.Scanner.throwFor(Unknown Source)

at java.util.Scanner.next(Unknown Source)

at Main.main(Main.java:414)

Figure 1.0

## **Final Award Calculation**

Enter Student Name and Id:-

chamod

2015300

---Level 04---

Module 1:-

ICT1 : 54

ICT2 : 95

ICT3 : 87

Module 2:-

ICT1 : 67

ICT2 : 94

ICT3 : 67

Module 3:-

ICT1 : 54

ICT2 : 6

ICT3 : 5

Module 4:-

ICT1 : 54

ICT2 : 97

ICT3 : 54

Module 5:-

ICT1 : 67

ICT2 : 88

ICT3 : 57

Module 6:-

ICT1 : 87

ICT2 : 79

ICT3 : 89

Module 1:-

Average = 78

PASS

Module 2:-

Average = 76

PASS

Module 3:-

Average = 21

RETAK

Module 4:-

Average = 68

PASS

Module 5:-

Average = 70

PASS

Module 6:-

Average = 85

PASS

Credits = 100

You are eligible for move to the Level-05 but you have to RETAKE your failure Module

Choose Your Career:-

(1)Continue

(2)Exit with CerHE

1

---Level 05---

Module 1:-

ICT1 : 65

ICT2 : 78

ICT3 : 97

Module 2:-

ICT1 : 67

ICT2 : 97

ICT3 : 73

Module 3:-

ICT1 : 67

ICT2 : 64

ICT3 : 97

Module 4:-

ICT1 : 64

ICT2 : 94

ICT3 : 99

Module 5:-

ICT1 : 49

ICT2 : 100

ICT3 : 100

Module 6:-

ICT1 : 94

ICT2 : 87

ICT3 : 56

Leve 04 RETAKE Module :-

ICT1 : 78

ICT2 : 88

ICT3 : 69

Module 1:-

Average = 80

PASS

Module 2:-

Average = 79

PASS

Module 3:-

Average = 76

PASS

Module 4:-

Average = 85

PASS

Module 5:-

Average = 83

PASS

Module 6:-

Average = 79

PASS

Level 04 RETAKE Module:-

Average = 40

PASS

Credits = 140

You are eligible for move to the Level-06  
and you passed the RETAKE module of Level-04

Choose Your Career:-

(1)Continue

(2)Exit with DipHE

1

```

---Level 06---

Module 1:-
ICT1 : 87
ICT2 : 96
ICT3 : 57
Module 2:-
ICT1 : 65
ICT2 : 78
ICT3 : 67
Module 3:-
ICT1 : 59
ICT2 : 79
ICT3 : 57
Module 4:-
ICT1 : 69
ICT2 : 99
ICT3 : 87
Module 5:-
ICT1 : 87
ICT2 : 85
ICT3 : 64

Module 1:-
Average = 80
PASS
Module 2:-
Average = 70
PASS
Module 3:-
Average = 65
PASS
Module 4:-
Average = 85
PASS
Module 5:-
Average = 78
PASS

Credits = 120
You are eligible for the HONORS DEGREE

-----
|Total Credits = 360                               |
|Total Average = 79                                 |
|Class:-1st Class Honors                           |
|Degree:-B.Sc. 1st Class Honors DEGREE             |
-----

Choose Your desire:-
(1)Add a Student
(2)Finish

```

Figure 1.1

## Final Report

Student Name :-chamod		Student Id :-2015300
Level-04	Level-05	Level-06
Module 01:-78 PASS	Module 01:-80 PASS	Module 01:-80 PASS
Module 02:-76 PASS	Module 02:-79 PASS	Module 02:-70 PASS
Module 03:-21 RETAKE	Module 03:-76 PASS	Module 03:-65 PASS
Module 04:-68 PASS	Module 04:-85 PASS	Module 04:-85 PASS
Module 05:-70 PASS	Module 05:-83 PASS	Module 05:-78 PASS
Module 06:-85 PASS	Module 06:-79 PASS	
Re Module:-40 PASS	null null null	
Total		
Credits		
=		
360		
Total		
Average		
=		
79		
Class:-1st		
Class		
Honors		
Degree:-B.Sc.		
1st		
Class		
Honors		
DEGREE		

Figure 1.2

## White Box Testing

CONDITION	YES	NO
If (marks1<=100&&marks2<=100&& marks3<=100)	Move to the next condition.	Display “invalid”
If(marks1<40  marks2<40   marks3<40)	Display “Condoned”	Move to the next condition.
If ( module average>=40 &&(marks1<40  marks2<40   marks3<40))	Display “pass” , “Condoned”	Move to the next condition.
If module average>=40	Display “Pass”	Move to the next condition.
If module average>=30	Display “Resit” and “condoned”	Move to the next condition.
If module average<30	Display “Retake”	-.
If module average>=30	Credits= Credits+20	Move to the next condition.
If credits1/2/3==120	Can move to the next level.	Can’t move to the next level.
If credits1/2/3==100	Can move to the next level. Display “you have to Retake your failure module”	Can’t move to the next level
If final credits<100	Can’t move to the next level.	-.

If cerhe/diphe= =1	Continue to next level	Exit with CerHE/Dip HE -.
If total credits==360	Display "Honors degree"	Move to the next condition.
If total average marks>=70	Display "1 <sup>st</sup> class honors degree"	Move to the next condition.
If total average marks>=60	Display "2 <sup>nd</sup> class honors upper division degree"	Move to the next condition.
If total average marks>=50	Display "2 <sup>nd</sup> class honors lower division degree"	Move to the next condition.
If total average marks>=40	Display "3 <sup>rd</sup> class honors degree"	Display "No class "
While num= =1	Add a student	End the program .

Table 1.2

## Conclusion

I learnt how to code a CLI based program and also I learnt lot of things about class and objects ,generate a text file and how to write to the file and read from the file.