The New Core Programming Curriculum at ICT

The School of ICT has completed the rollout of its new core programming curriculum for its CPA, CPAC, CPD and BSD programs. This report presents the results of this rollout. The results combine the data collected on student success along with recommendations of faculty who have taught the update subjects as of the end of the Fall Semester of 2014.

Faculty conveyed results on the current state of the rollout to the author, who collated them and presented the summary version to a face-to-face meeting on December 17 2014. At that meeting faculty members reached consensus on nine recommendations. These initiatives are listed in the penultimate section of this report.

Background

The upgrade to our core programming curriculum has taken 4 semesters to complete. We initially offered our new 1st semester courses – IPC144 and BTP100 – in Fall 2013, our new 2nd semester courses – OOP244 and BTP200 – in Winter 2014, and our new 3rd semester courses – OOP345 and BTP305 – in Fall 2014.

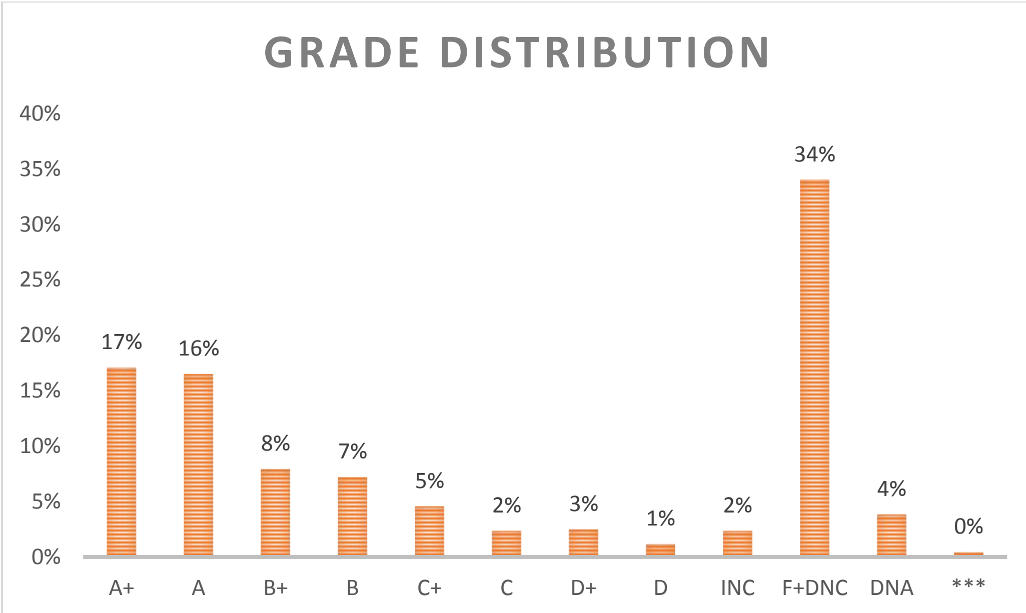
The mid-semester report dated November 2014 [1] detailed a strategy for shifting from our conventional teaching and learning framework to one in which students are active partners in the teaching and learning process. That report identified strong correlations between workshop completion rates and student success rates in both our diploma and degree programs.

The mid-semester report also predicted an average pass rate of about 65%. This pass rate was notably higher than the ~50% pass rate under our conventional teaching and learning framework, based on anecdotal evidence. Seneca College's Academic Plan for 2012-2017 defined an 80% 1st-semester pass rate target with a 70% graduation pass rate target. In other words, although our experiential learning implementation has proved positive with respect to our own experience, this new pass-rate falls well short of the 80% target defined in the Academic Plan. Clearly, further initiatives are needed to approach this ambitious target.

Observations

The data collected from faculty covers 679 students in our core programming subjects: 540 in our diploma programs (CPA, CPAC, and CPD) and 139 in our BSD program.

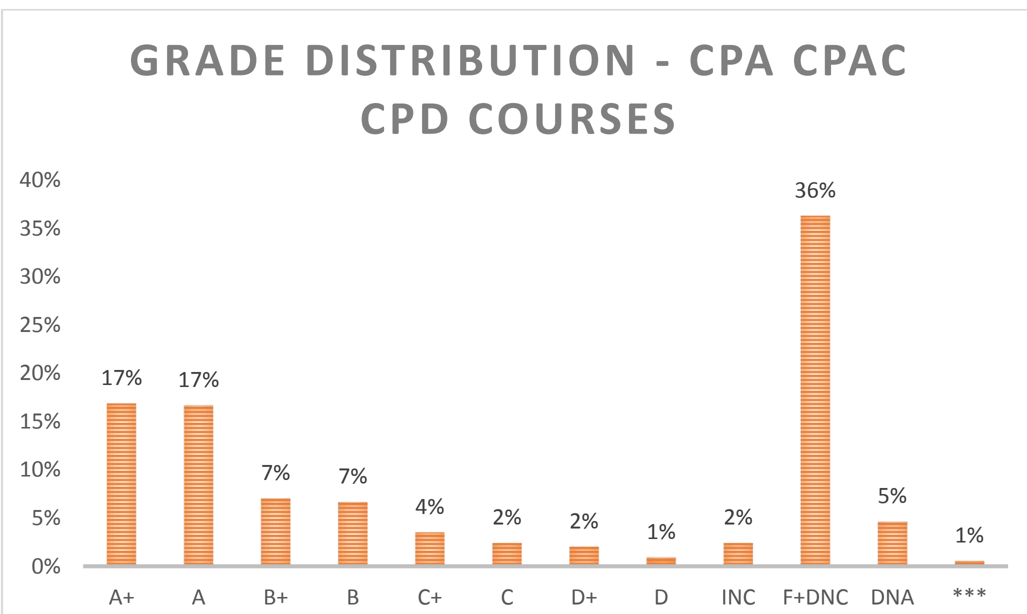
The grade distribution for all core programming subjects is illustrated in Figure 1. One third of our students was highly successful (A, A+); a third was unsuccessful (F + DNC).



Grade Distribution for All Core Programming Subjects

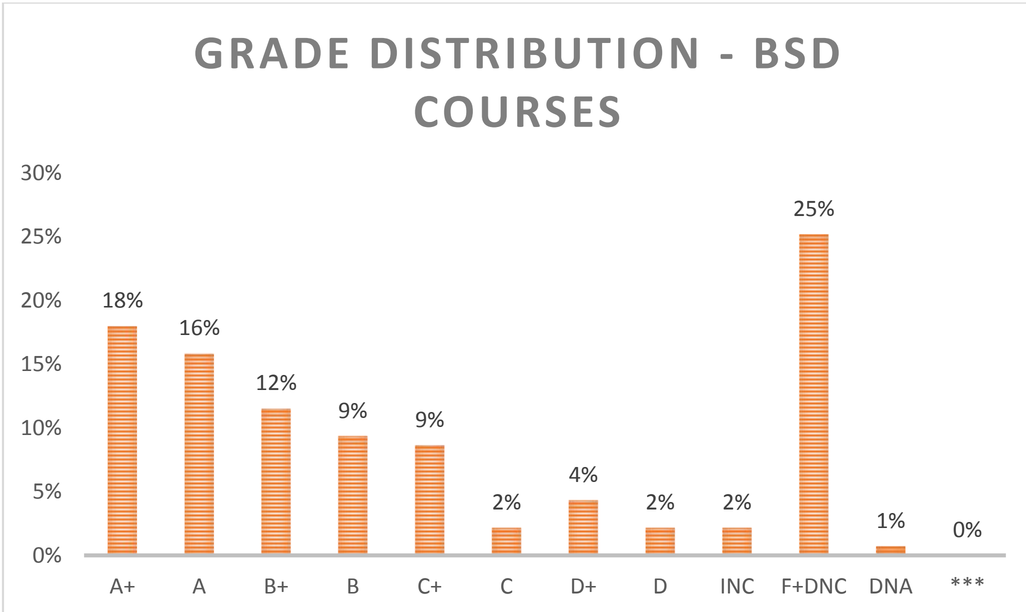
Figure 1

Separate distributions for the diploma and degree programs are illustrated in Figures 2 and 3 respectively. Although the same proportions of students were highly successful in each set, a significantly higher proportion was unsuccessful in the diploma set: 36% as opposed to 25% in the degree program.



Grade Distribution for All Diploma Core Programming Subjects

Figure 2



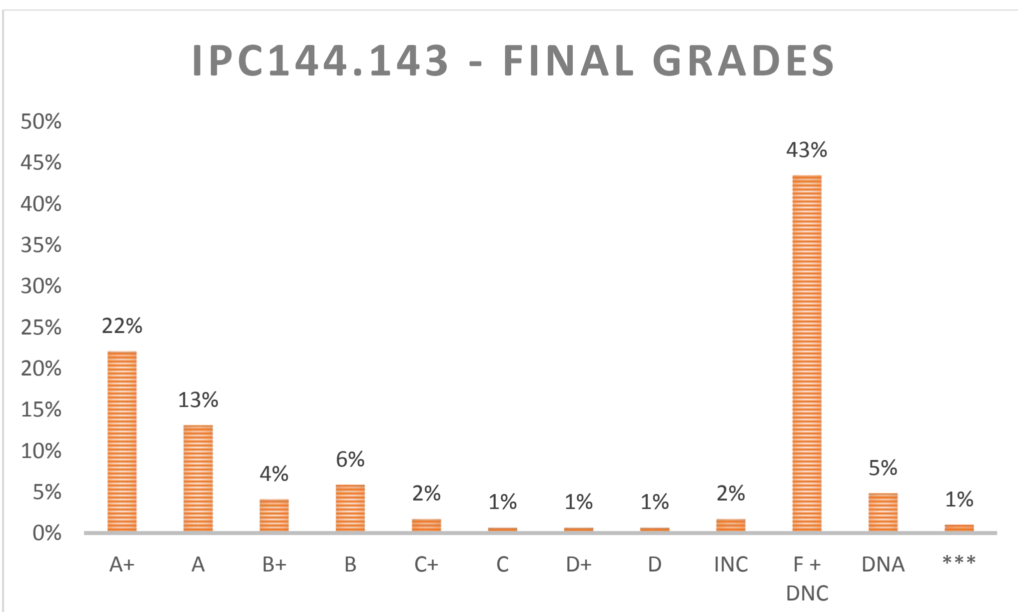
Grade Distribution for All Degree Core Programming Subjects

Figure 3

Diploma Programs – Per-Subject Distributions

The grade distributions for the diploma courses are illustrated in Figures 4 through 6.

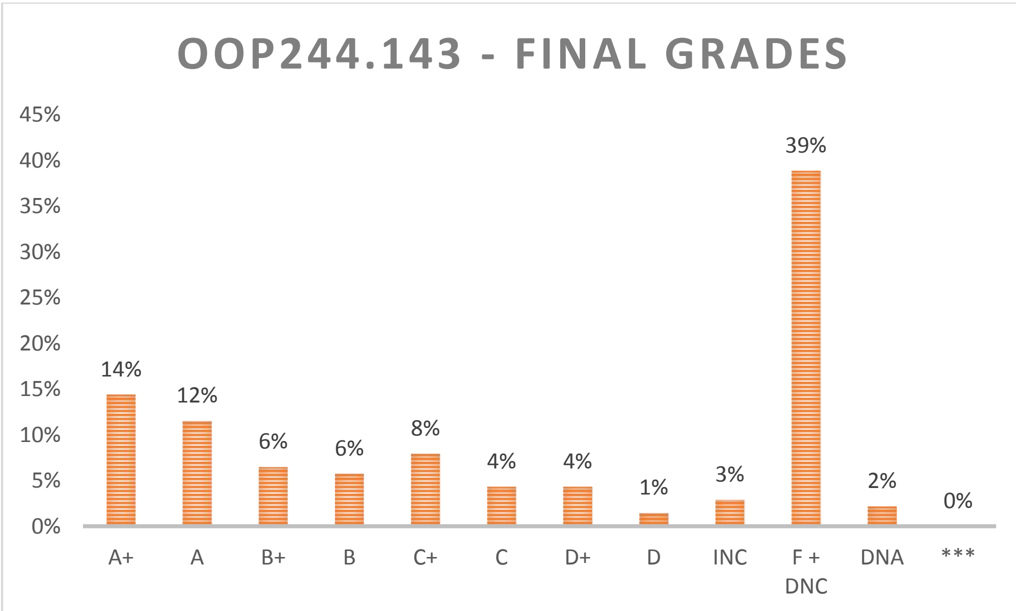
Six instructors taught 290 IPC144 students in 11 sections. Few students achieved intermediate results.



Grade Distribution for IPC144

Figure 4

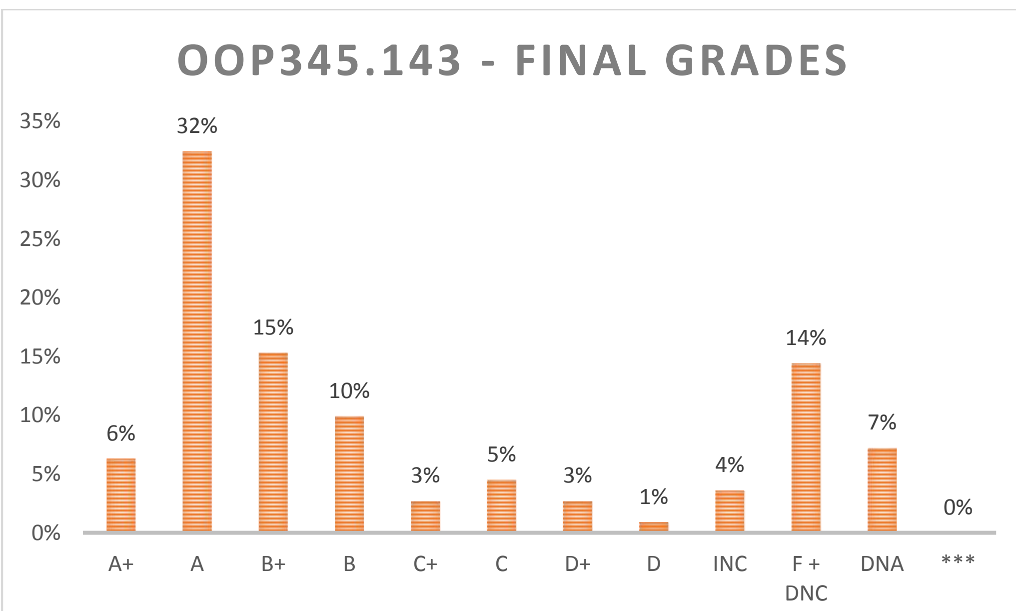
Two instructors taught 139 OOP244 students in 4 sections. A third of the students achieved intermediate results.



Grade Distribution for OOP244

Figure 5

Two instructors taught 111 OOP345 students in 4 sections. Well over a third of the students achieved intermediate results.



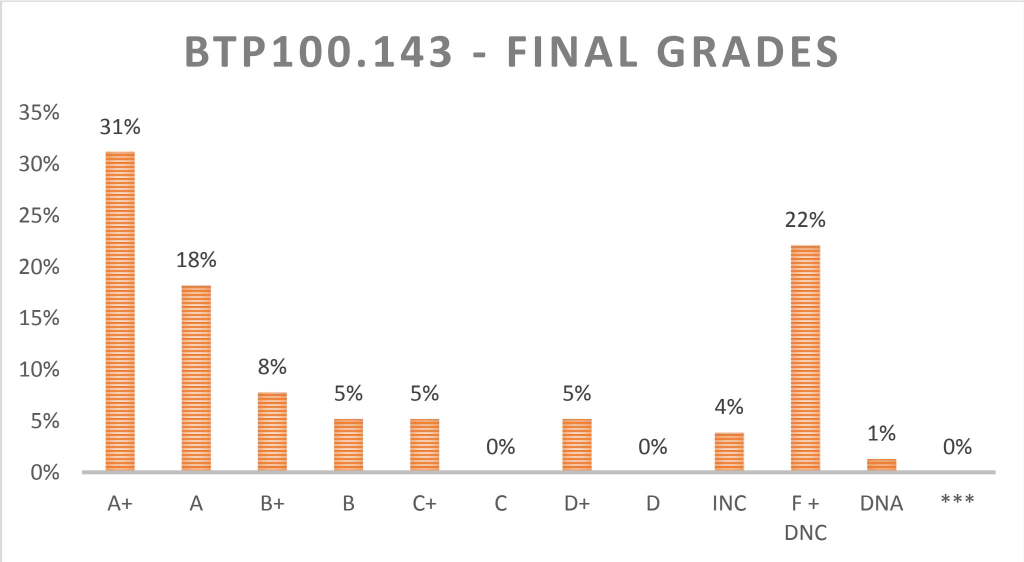
Grade Distribution for OOP345

Figure 6

Degree Program – Per-Subject Distributions

The grade distributions for the degree courses are illustrated in Figures 7 and 8.

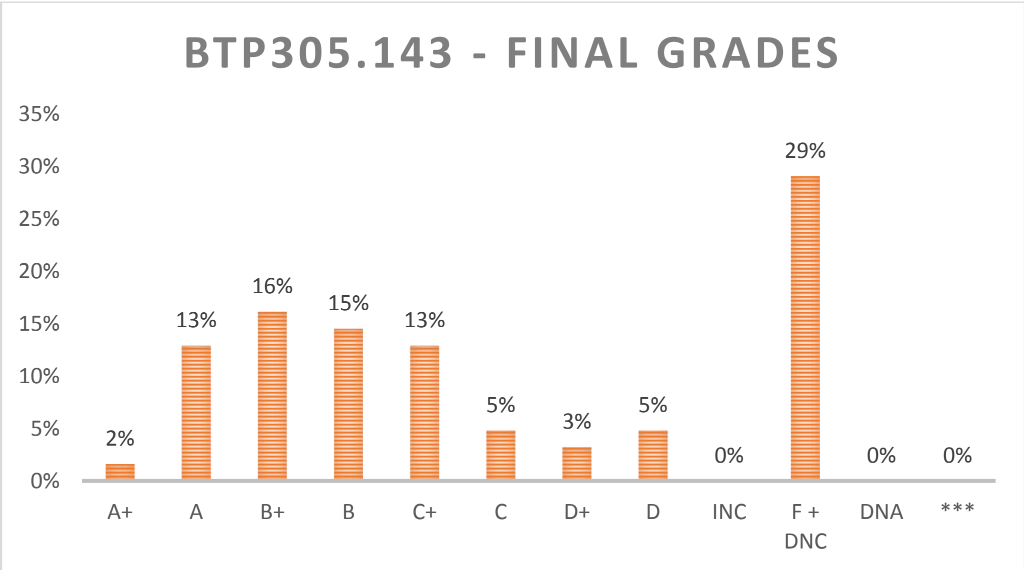
A single instructor taught 77 BTP100 students in 2 sections. Under one third of the students achieved intermediate results.



Grade Distribution for BTP100

Figure 7

A single instructor taught 62 BTP305 students in 2 sections. Well over half the students achieved intermediate results.



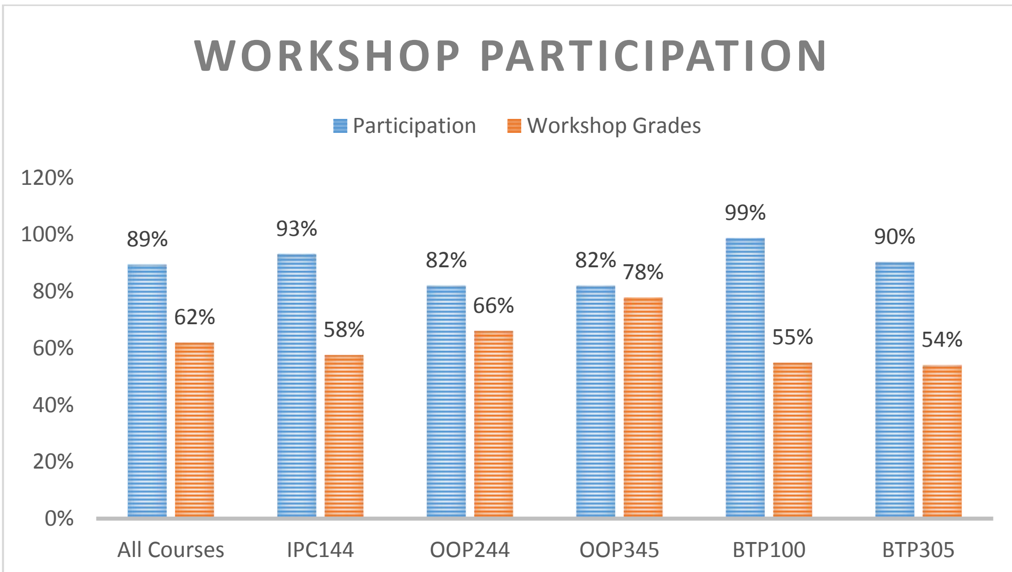
Grade Distribution for BTP305

Figure 8

Workshop Participation

The workshop participation rate across all core programming subjects was quite high (89%). This rate refers to the average number of submissions regardless of quality. The overall quality rate was significantly lower (62%). That rate refers to the actual grades earned for the work submitted by those who participated.

The workshop participation and quality rates for each subject are illustrated in Figure 9. The far left pair of bars shows the rates for all core programming subjects combined.



Workshop Participation and Quality Rates

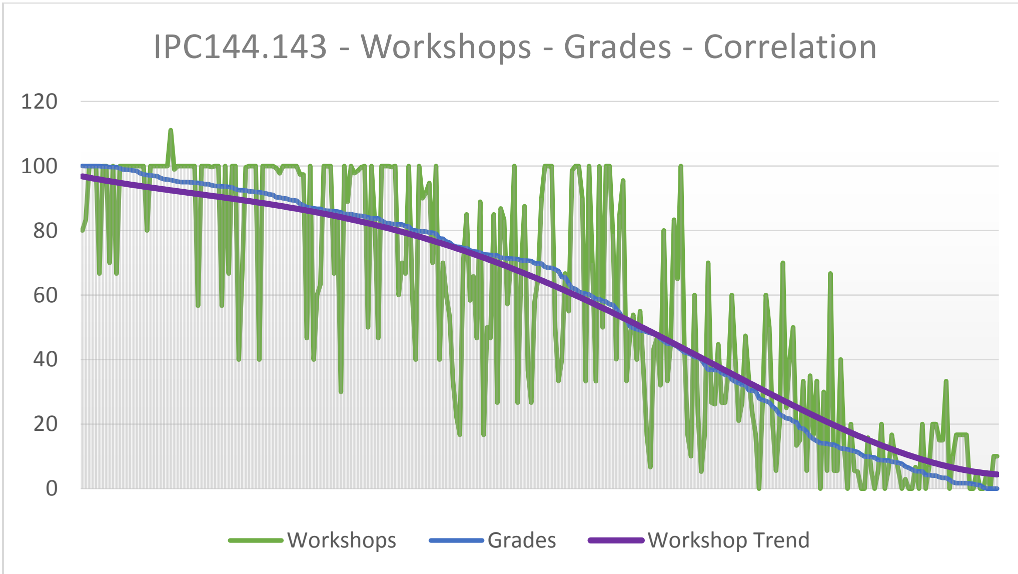
Figure 9

The quality of workshop submissions was lowest in the 1st semester for both diploma and degree programs. Since no data was collected with respect to assessment criteria for each subject it is difficult to draw further conclusions. The workshop assessment policies and weights varied between instructors and the overall results on quality aren't very reliable.

Workshop Quality-Student Success Correlations

The workshop quality correlated well with student success in all subjects. Students who completed all of their workshops earned a passing grade, while those who completed less than half or submitted lower quality work were often unsuccessful.

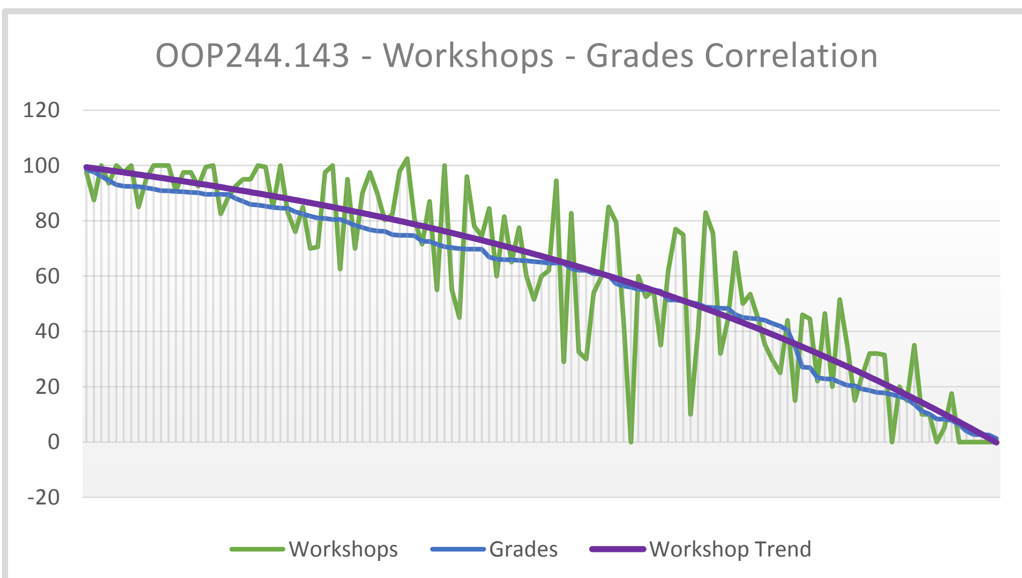
The weighting for workshop submissions varied between instructors (15% to 20%) – some placed more weight on quizzes while other instructors placed more weight on the workshops and less on quizzes. The correlation between workshop quality and student success in IPC144 is illustrated in Figure 10.



Workshop Quality Student Success Correlation for IPC144

Figure 10

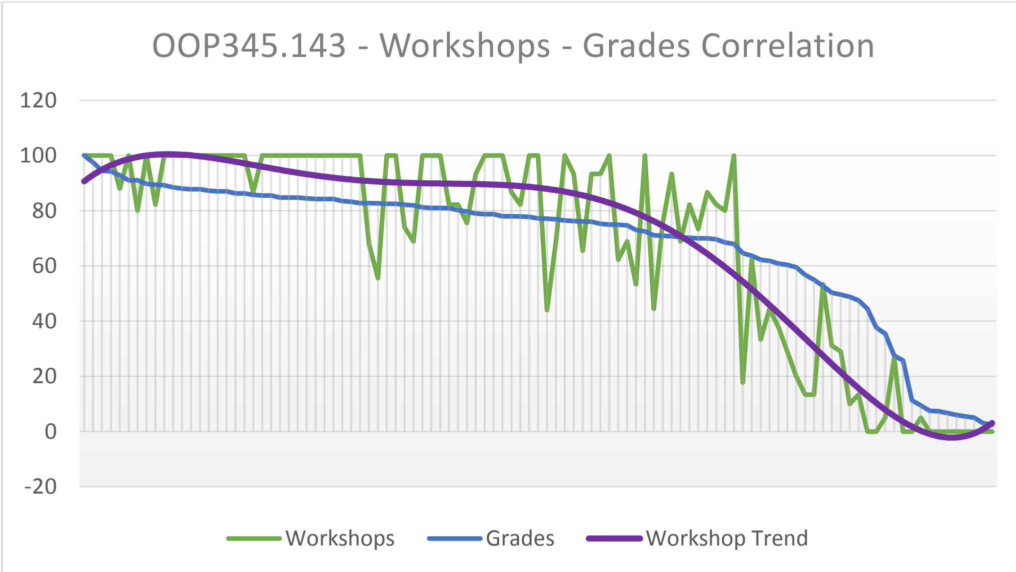
The weighting of workshops in OOP244 was 20% for both instructors. The variation about the trendline was notably smaller than for IPC144. The correlation between workshop quality and student success in OOP244 is illustrated in Figure 11.



Workshop Quality Student Success Correlation for OOP244

Figure 11

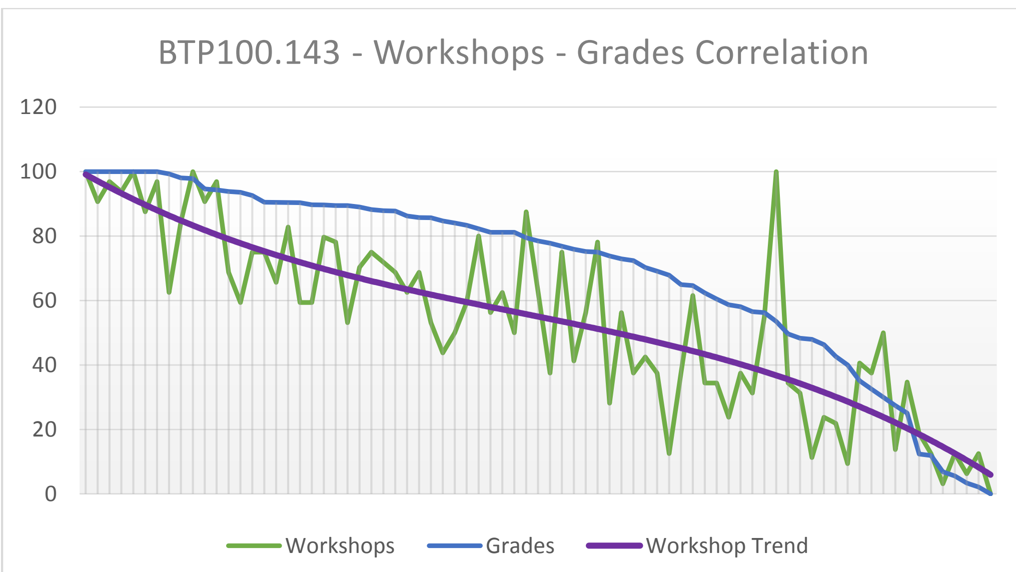
The weighting of workshops in OOP345 was 20% for both instructors. The variation about the trendline was notably smaller than for IPC144. The correlation between workshop quality and student success in OOP345 is illustrated in Figure 12.



Workshop Quality Student Success Correlation for OOP345

Figure 12

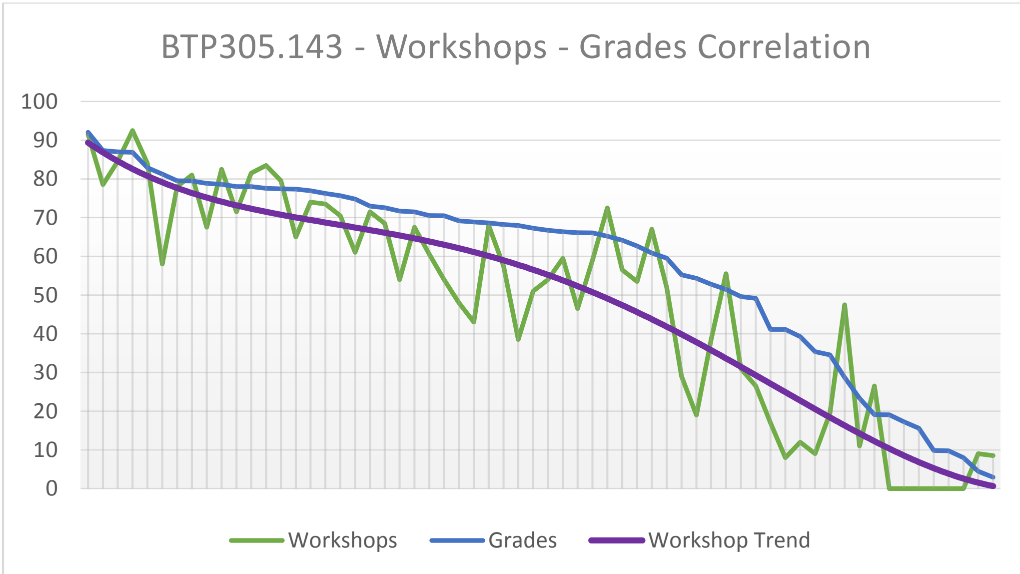
The weighting of workshops in BTP100 was only 9%. The trendline lies further from the grades line than in those subjects where the weighting was higher. The correlation between workshop quality and student success in BTP100 is illustrated in Figure 13.



Workshop Quality Student Success Correlation for BTP100

Figure 13

The weighting of workshops in BTP305 was 20%. The variation about the trendline was notably smaller than for IPC144. The trendline lies slightly closer to the grade line than for BTP100 students.



Workshop Quality Student Success Correlation for BTP305

Figure 13

Analysis

The experiential learning results confirm a strong correlation between workshop quality and student success. The results for IPC144 are quite scattered, while the results for OOP244 and OOP345 are more regular and more in line with the grades distribution. The results for both BTP100 and BTP305 are regular but do lie well below the grades line.

The grade distributions themselves show an obvious weakness in the middle third. We can address this weakness through two distinct approaches. We can:

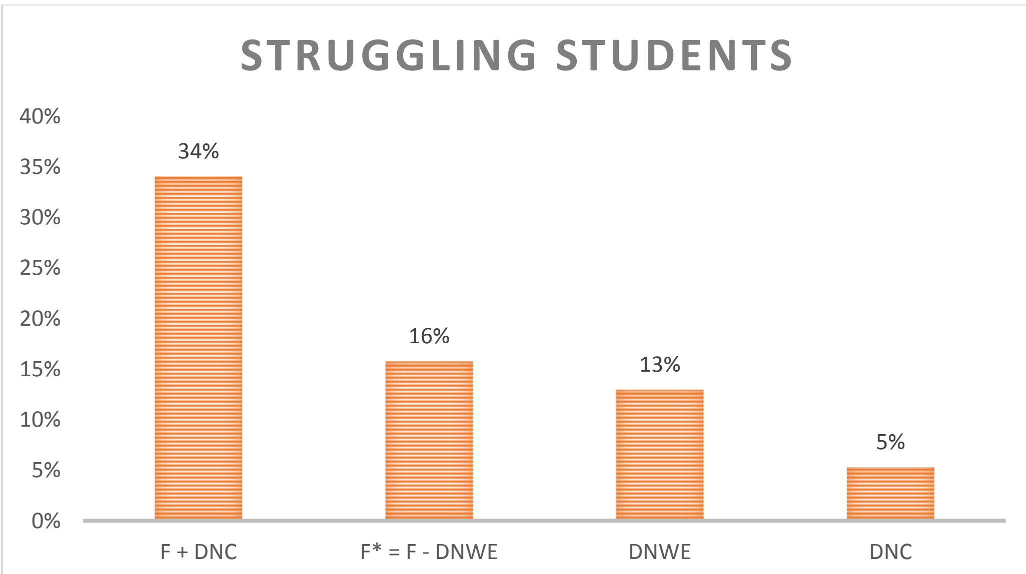
a) identify the learning barriers separating highly successful, intermediate and struggling students and propose appropriate strategies for each student type
b) distinguish the struggling students according to categories of awareness

For the first approach we can turn to theories of cognition tested on our programming students. Experimental results on Seneca students dealing with ill-structured problems, similar to those encountered in programming, are reported in [2]. The barriers between novice, intermediate and advanced learners are investigated in that paper. Formulating and presenting problems specialized for each student type may prove worthwhile.

For the second approach, let us identify the different categories of awareness that flow from our observations:

1) DNC – did not complete and dropped the subject before the drop date (these students presumably realized before the drop date that they do need to retake the subject)
2) DNWE – did not write the exam (these students presumably realized after the drop date but before the exam that they could not pass the subject)
3) F* = F – DNWE – wrote the exam but did not earn enough marks to pass the subject (these students failed the exam or did not earn enough grades to pass the subject)

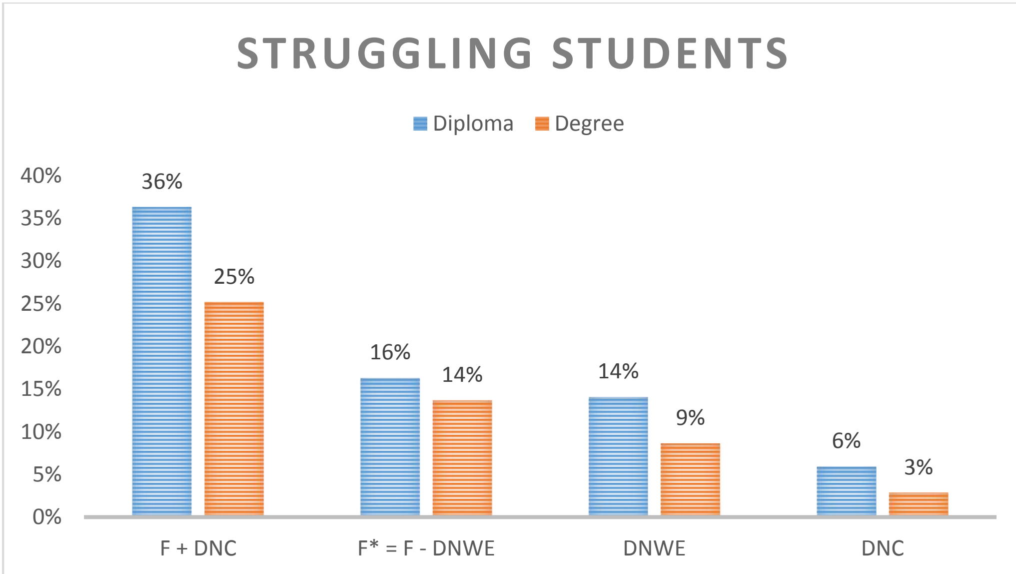
An analysis of the grade distributions for all students in our core programming subjects into these categories of awareness is illustrated in Figure 14. The bar on the far left shows the sum of the three categories.



Analysis of Struggling Students into Awareness Categories

Figure 14

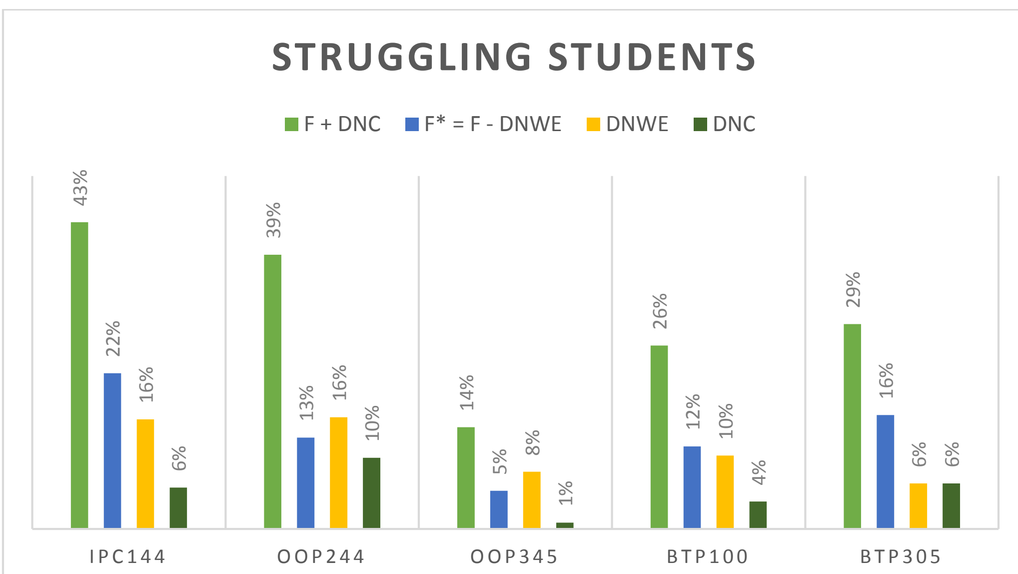
A comparison of this analyses between our diploma and degree students is illustrated in Figure 15. This comparison shows that the degree students performed better in each category than did the diploma students.



Comparison of Struggling Diploma and Degree Students

Figure 15

A comparison of the analyses across the different subjects is illustrated in Figure 16. The complete grade distribution for each subject using these additional categories is illustrated in the Appendix. This comparison shows that different strategies for each category may be warranted.



Comparison of Struggling Students by Subject

Figure 16

In IPC144, 22% of the students demonstrated some hope that they might pass but were unsuccessful in the end, while 22% gave up before the final exam.

In OOP244, 13% of the students demonstrated some hope that they might pass but were unsuccessful in the end, while 26% gave up before the final exam.

In OOP345, 5% of the students demonstrated some hope that they might pass but were unsuccessful in the end, while 9% gave up before the final exam.

In BTP100, 12% of the students demonstrated some hope that they might pass but were unsuccessful in the end, while 10% gave up before the final exam.

In BTP305, 16% of the students demonstrated some hope that they might pass but were unsuccessful in the end, while 12% gave up before the final exam.

Recommendations

Faculty Discussion

The faculty members who have taught the new curriculum met with other programming faculty on December 17 2014 to discuss their experiences and make recommendations. The list of faculty members who attended this meeting is in the appendices. The author reviewed the results described above at the start of this faculty meeting.

The faculty members made the following recommendations:

1) Edit and clarify word problem descriptions on both tests and exams. Improve conciseness and remove ambiguity for students who are still in initial stages of learning the English language. Implement a standard template with which the students become familiar throughout the semester so that they spend minimum time determining what is being asked of them. The focus in semesters one and two should be on testing their programming skills only. Leave the requirements analysis to more dedicated courses that run in parallel or follow afterwards.
2) Provide comprehensive and prompt feedback for each workshop submission in order to engage students. Hire part-time tutors for this purpose, leaving only the marking to the instructor. Tutor feedback should be available throughout the first three semesters.
3) Hold review sessions outside class time for students who have found material too difficult to understand the first time around. Part-time tutors can run these review sessions on a weekly basis.
4) Several faculty member expressed disappointment with the quality of service that learning center tutors provide. Comments included a lack of understanding of the more modern material being introduced and providing solutions instead of showing students how to find those solutions.
5) Prefer two double-sided reference sheets to open book during tests and exams.
6) Print test and exam papers in color. Students have found this particularly helpful in the reading of walkthrough problems.

7) Learn applications of Cognitive Load Theory to the teaching of our programming subjects. The material covered in the Foundations of Teaching and Learning is insufficient for programming students. Faculty expressed an interest in a deeper understanding of cognitive psychology.
8) Instructors need to step up to prepare labs for more variety beyond the default labs that are currently available. Several faculty members remarked that they do not have time to do so.
9) Weekly coordination meetings in multi-section subjects are quite important. They allow instructors to share experience, help identify student difficulties across the sections, ensure that instructors are on the same page with respect to emphasis and provide more equitable assessments.

Fardad noted that in order to accommodate active learning by his students, he prepared all of his examples before class and would move them to the screen as he lectured. This saved considerable time over entering code which requires the students to wait for him to finish. Elliott noted that he too prepared his sample code off-line for the same purpose.

Mary Lynn suggested holding a mid-semester promotion meeting for all first semester courses. Although beyond the scope of this report such a promotion meeting would flag those students struggling across all first semester subjects and may warrant a different kind of intervention than an individual subject intervention.

Author's Input

The following topics flow from the results presented here but were not discussed at the meeting to any particular conclusion. Recommendations of the mid-semester report are appended to this list:

a) Increase the weight of the workshops in BTP100 to 20%
b) Normalize the weight of the workshops in IPC144 at 20%
c) Turn the workshops into mini-assignments with one final project that integrates them as the sole assignment in each course
d) Stress the importance of attendance – workshop periods are not optional
e) Assist students who complete exercises and workshops yet fail their tests
f) Embed in-class exercises within lectures – instructor circulates amongst students [MLM - what about doing this as group or team in-class exercises. No more than 3 in a team. Reduces load on faculty and at the same time allows students to learn from each other. Ask a team to present their solution or many teams (e.g. Active Learning Classrooms are perfect for this(!) and then talk to a couple of teams' proposed solution to get a sense of why some solutions are better than others.]
g) Review and refine the learning outcomes for the 1st and 2nd semester courses

h) Identify more optional sections in the subject web sites and in-house textbooks to reduce content without interrupting flow through the central core of the material
i) Develop baskets of workshops for the courses to increase variety

As an afterword, the following set of questions comes to mind:

a) Learning Outcomes - Should the criteria for passing IPC144/BTP100 be defined primarily in terms of what is necessary to commence studying OOP244/BTP200 and those for passing OOP244/BTP200 be defined in terms of what is necessary to commence studying OOP345/BTP305?
b) Should the core programming subjects move towards more practice and away from the final exam making it optional (so students can improve their grade)?
c) Should we modify our policy of granting SUPs to include those weaker students who have completed the exercises, demonstrated effort throughout the course and only failed the final exam?

These are simply suggestions and still require future review and evaluation by faculty members who teach in our core programming subjects.

Concluding Remarks

The implementation of experiential learning in our core programming subjects appears to be well on its road to success. No evidence in the data presented here suggests that we return to a conventional and traditional lecture only mode of teaching.

This report confirms that workshop quality rates correlate well with student success rates for our core programming subjects and that workshops alone are insufficient to reach the College's target of an 80% pass rate in the 1st semester. This outcome was predicted in the mid-semester report. Further initiatives are certainly required if we are to reach close to this ambitious target. The present report includes recommendations from faculty who have recently taught our core programming subjects. Faculty reached a consensus on these initiatives. Faculty teaching the subjects in future should review these proposals and implement what they can where they can.

The suggestions here not yet agreed by faculty are speculative and open to experiment on an individual basis. Results from such initiatives could provide some basis for future debate on improvements in our curriculum.

Hopefully, this end-semester report provides context for further collaboration amongst our programming faculty in improving the learning environment for ICT students.

Chris Szalwinski, P.Eng., Ph.D.

Programming Curriculum Coordinator

School of Information and Communications Technology

December 2014

References

1 – Szalwinski, C.M. (2014). “Progress on Seneca’s Academic Plan Core Programming Subjects – School of ICT”. Internal Report, November 2014.

2 – Anastasiade, J. and Szalwinski, C.M. (2010). “Building Computer-Based Tutors to Help Learners Solve Ill-Structured Problems”. In Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications 2010. Toronto, Ontario, Canada: Association for the Advancement of Computing in Education. pp. 3726–3732.

Appendices

Faculty who attended the December 17 2014 Meeting

Greg Blair

Elliott Coleshill

Elnaz Delpisheh

Justin Denney

Mark Fernandes

Joseph Hughes

Hasan Kamal Al-Deen

Cathy Leung

Franz Newland

Oliver St-Cyr

Yue (Sunny) Shi

Marcel Silva

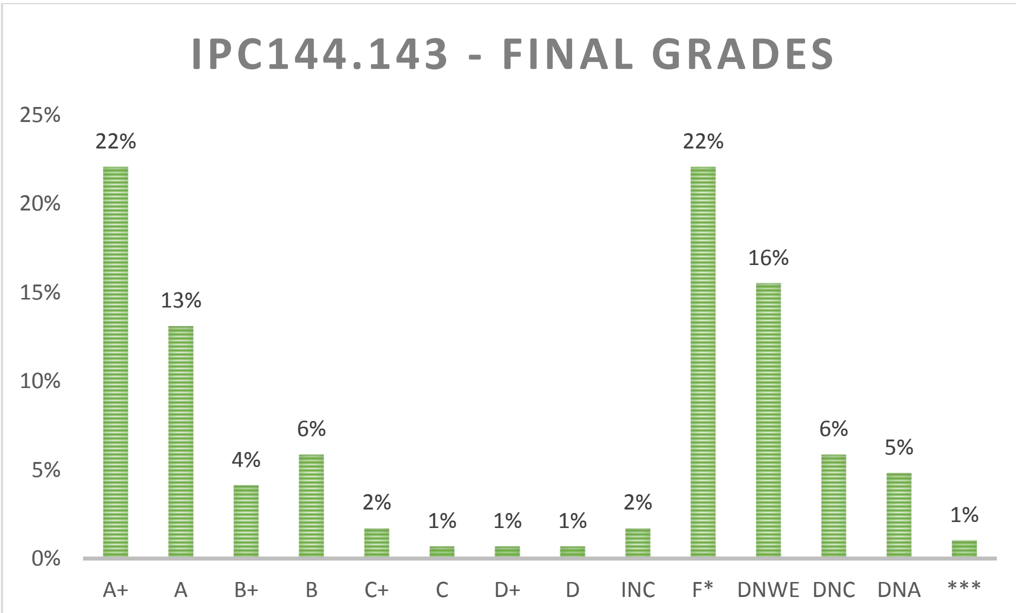
Fardad Soleimanloo

Chris Szalwinski

Ian Tipson

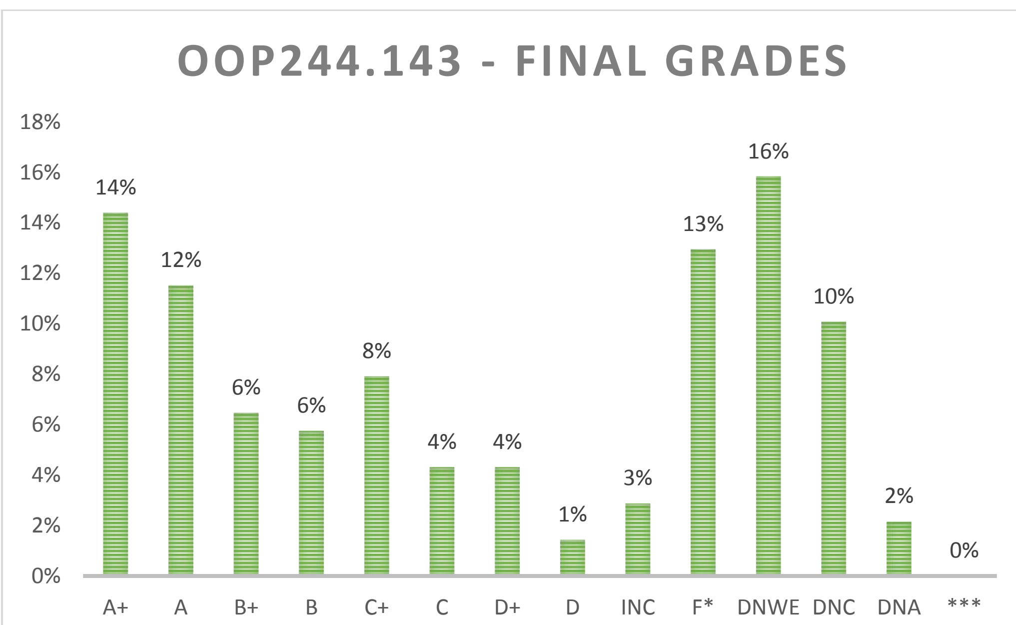
Appendices

Refined Grade Distributions by Subject



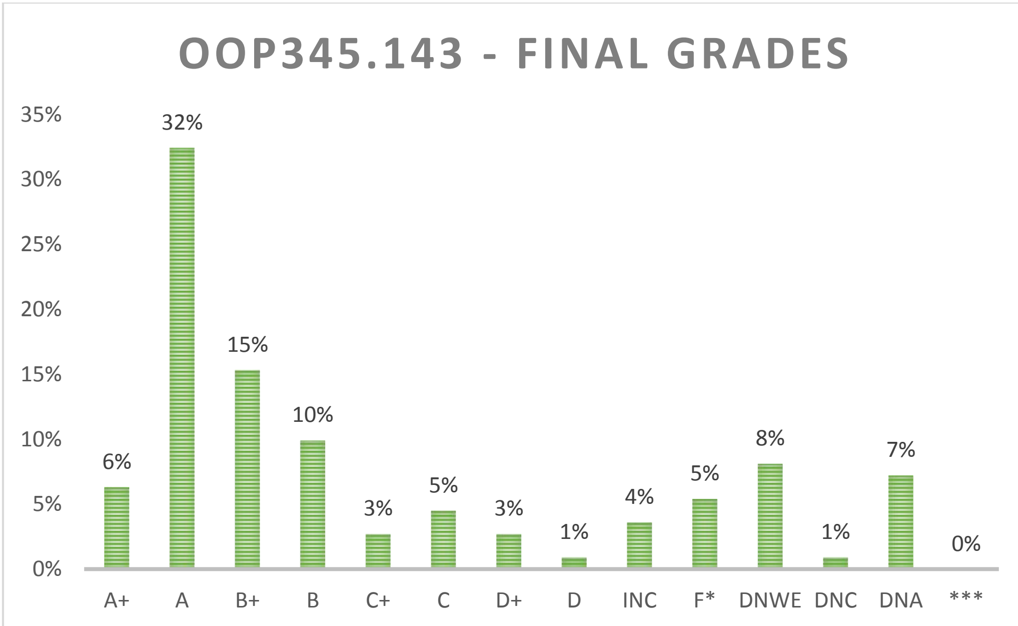
Refined Grade Distribution for IPC144

Figure A1



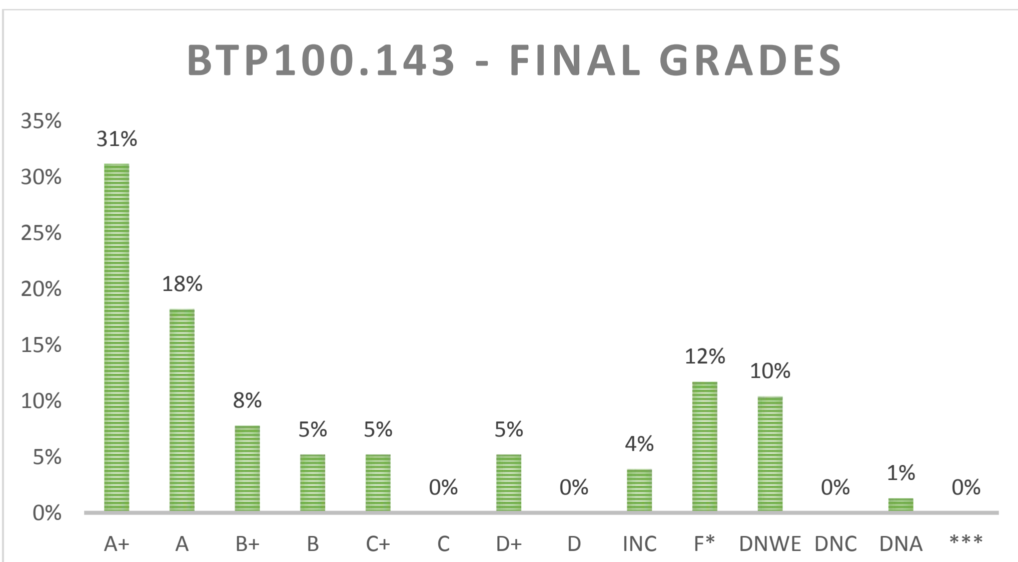
Refined Grade Distribution for OOP244

Figure A2



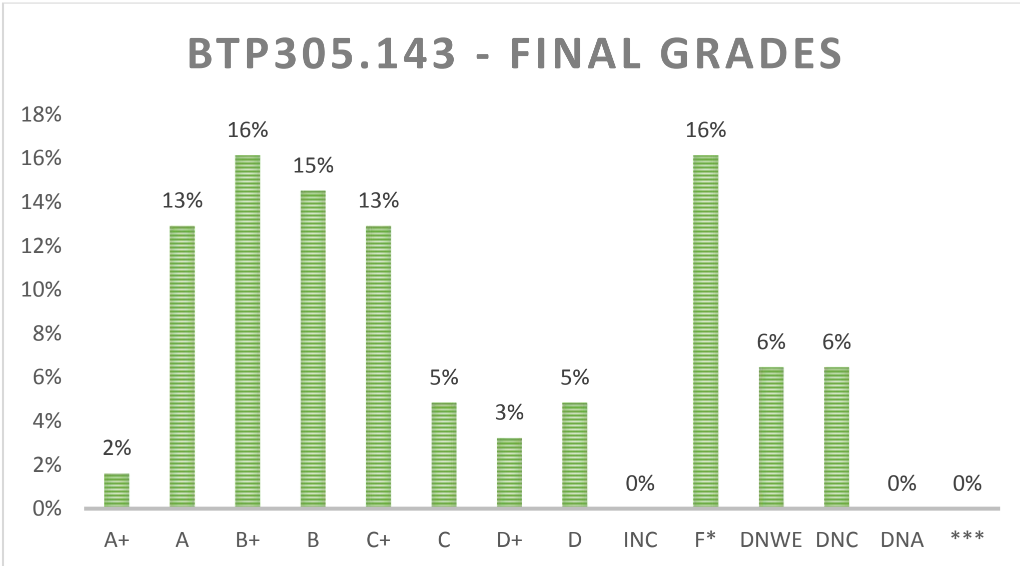
Refined Grade Distribution for OOP345

Figure A3



Refined Grade Distribution for BTP100

Figure A4



Refined Grade Distribution for BTP305

Figure A5