2021 ASEE ANNUAL CONFERENCE
Virtual Meeting | July 26–29, 2021 | Pacific Daylight Time
Paper ID #34947

# Pandemic Response: Hybrid-flexible Course Delivery for General Education Computer Science Courses

**Mr. Ben Ralph Bernard, North Dakota State University**

Ben Bernard is a Doctoral Candidate in the Department of Computer Science at North Dakota State University and is the Computer Services Specialist for the Department of Architecture & Landscape Architecture at NDSU with sixteen years of service. Bernard has been professionally recognized for his contributions to NDSU's new cybersecurity educational programs and is the first to be awarded an MS in Computer Science with an Cybersecurity focus at NDSU. His research interests bring new technologies together with creative education and outreach implementations, frequently featuring digital fabrication, augmented & virtual reality, and cybersecurity education. Bernard leads a team of graduate & undergraduate design students to research and implement advanced design technologies.

**Dr. Jeremy Straub, North Dakota State University**

Jeremy Straub is the Director of the NDSU Institute for Cyber Security Education and Research and an Assistant Professor in the Department of Computer Science at the North Dakota State University. He holds a Ph.D. in Scientific Computing, an M.S. and an M.B.A. and has published over 40 journal articles and over 120 full conference papers, in addition to making numerous other conference presentations. Straub's research spans the gauntlet between technology, commercialization and technology policy. In particular, his research has recently focused on cybersecurity topics including intrusion detection and forensics, robotic command and control, aerospace command and 3D printing quality assurance. Straub is a member of Sigma Xi, the AAAS, the AIAA and several other technical societies, he has also served as a track or session chair for numerous conferences.

**Pandemic Response: Hybrid-Flexible Course Delivery for General Education Computer Science Courses**

## Abstract

The COVID-19 pandemic has sparked significant interest in the HyFlex course model. It allows colleges and universities to temporarily close or reduce in-person attendance and apply social distancing policies to classrooms to reduce the risk of disease spread among the student and faculty populations. Thus, it enables students and instructors who prefer instruction in a physical classroom the opportunity to safely do so.

This paper covers the adaptation of a course, CSCI 159: Computer Science Problem Solving, to the HyFlex model. This paper presents the strategies used and the lessons learned about how to effectively instruct general education computer sciences courses using the HyFlex course model.

## Introduction

The HyFlex course delivery model was originally developed to help increase enrollment by allowing students to participate in a class without being physically present in a classroom [1]. It has become widely implemented in the wake of the coronavirus pandemic because of the flexibility it offers both students and faculty. Using the HyFlex model, instructors build content for both a fully online course and a 'face-to-face' classroom environment. The HyFlex course uses both learning environments in tandem, sharing the same learning outcomes each week so that students can move between online and in-person experiences in the course seamlessly.

CSCI 159: Computer Science Problem Solving was identified as a course to be offered via online and then online-based HyFlex models to reach students who could not enroll in traditional classroom sections of the course. It was first introduced as such in the fall 2019 semester. It also became NDSU's first STEM course to be offered for dual-credit enrollment to both high school and college students. CSCI 159 is a general education course that is designed to provide students an introduction to computer science and teach them how the computer science discipline applies quantitative reasoning to analyze data, create algorithms, and solve real world problems.

Building on the lessons learned in the first semester and the strengths of the HyFlex model, the course was well positioned to help students succeed in spite of the many challenges of the coronavirus pandemic in the spring and fall semesters of 2020.

## Background

In 2005, the Instructional Technologies graduate program at San Francisco State University was challenged with increasing enrollment and providing more participation opportunities for students [1]. Rather than adopt a learning model that would require students to choose between the traditional classroom environment and a pure online class, or a blended model in which the instructor chooses which course activities are placed in a physical classroom or online, Dr. Brian Beatty created and implemented a 'hybrid-flexible' learning environment. In this environment, students decide for themselves if they wish to participate in the course in the classroom (face-to-face) or online (synchronously or asynchronously), freely throughout the course [2]. Each class offered in this HyFlex delivery format has classroom, synchronous and asynchronous online content running simultaneously. Students choose to attend as many or as few of the classroom sessions as they want and complete the rest of the course online. The course objectives are structured so that students can complete them in the classroom or online; the instructor does not favor or require one course delivery format over the other.

The HyFlex model offers several advantages to both students and faculty [3]. Students receive increased access to courses and more control over how and when to access courses. Courses only offered in the traditional classroom model sometime create course conflicts in which a student is forced to choose one class over another. Students juggling the responsibilities of part-time or full-time employment may not be able to regularly attend a scheduled classroom course. Students also benefit from more learning resources and opportunities, as HyFlex allows students to review the online course content on demand while still being able to take advantage of face-to-face instruction.

Faculty get the opportunity to build experience teaching online without giving up traditional classroom instruction. They can serve more students with only a small increase in the resources needed. Students and faculty are both able to take advantage of the HyFlex model when absent from the classroom without falling behind in the course material – a key advantage during the COVID-19 pandemic [4].

Concerns about this model exist, however. Some faculty worry that providing the opportunity for students to complete the course materials online will result in significantly fewer students who choose to be present in the classroom. Some studies, though, have indicated a higher number of course materials provided online was associated with fewer student classroom absences [5]. Studies have also shown that students value recorded lectures, also known as 'lecture capture', have good awareness of the technology functionality, and use the recorded lectures both for reviewing course content and making up for classroom absences [6]. Making lecture slides available online prior to a classroom lecture can also improve classroom attendance [7].

**Design**

To correctly design and implement a HyFlex course, it is important to ensure that the student participation modes are meaningful, that all learning activities in the student participation modes lead to equivalent learning outcomes, and that students can choose how they wish to participate in the course freely [8]. If students do not have the flexibility to choose how they wish to complete the course activities in any given week, it becomes a standard 'hybrid' course in which students are 'locked in' to either the traditional 'face-to-face' classroom or the online version of the course at enrollment. Online-only HyFlex offerings maintain the flexibility of the HyFlex model without the classroom attendance option.

CSCI 159: Computer Science Problem Solving is a general education course offered at NDSU that satisfies the quantitative reasoning requirement [9]. It is described in the University Bulletin as a course where "computer-based problem solving techniques are introduced in the context of the Internet, including web-site development. Programming concepts, data structures and algorithms, as well as modeling techniques are discussed" [10]. To achieve the specified general education quantitative reasoning course outcomes [9], it was designed to provide beginning students an introduction to computer science and teach them how the computer science discipline "applies quantitative reasoning to analyze data, create algorithms, and solve real world problems" [11]. Students are introduced to information systems, networking, web site development, and how to write computer programs with Python. As a general education course, it is available to any student to use to complete the requirements for their degree. As a result, students enrolled in the course have diverse academic backgrounds and a wide range of prior experience with computer science concepts.

Course material is divided into weekly subjects ranging from an introduction to information technology and networking, creating a webpage using raw HTML, to an introduction to algorithms and programing concepts with Python. Each week, recorded lectures, course slides, and related online materials are provided via the university's online course management system (Blackboard). Classroom lectures can be streamed online (synchronous online learning) and are recorded for review later (asynchronous online learning). Recording lectures creates the opportunity to use student interaction time to answer more student questions on the course material. This concept is the basis of a classroom pedagogy commonly known as the 'online flipped classroom model' [12]. The flipped classroom model has previously been shown to be effective with teaching an introductory programing course using Python [13] and object-oriented programing and design with Java [14], [15].

Another advantage that recorded lectures offer is the ability for students to review lecture content at their own pace. Learners can participate in classroom lectures and activities more freely since they can take more detailed notes from the recorded lectures outside of the classroom. Without the option of a recorded lecture, some students may be concerned about their ability to 'keep up' with the pace that content is presented at the lecture by the instructor. Students can use the recorded lectures to review and study the more challenging content in the course.

Course resources and materials were specifically chosen to be platform independent and to perform well with limited bandwidth. All resources were also free for learners to use. PythonAnywhere provided a full Python environment in a web browser which students used to share their Python consoles with each other or their instructor [16]. The course instructor does not have to provide custom installation instructions for students using different operating systems or editors, which creates more time to teach programming concepts. Other examples of platform independent, low bandwidth course resources are "How to Think Like a Computer Scientist: Interactive Edition" by the Runestone Interactive Project [17] and Trinket.io's Hour of Python tutorials [18]. Course materials were chosen to take advantage of high quality open educational resources (OER) to reduce the costs for learners. NDSU's Student Government created a grant program to encourage faculty to use OER [19] and created a joint committee with the NDSU Libraries called the Open Resources Board (ORB) to advocate for the implementation and sustainability of open educational resources at both the university and state level [20]. This effort shows the perceived importance of using OER to students.

The weekly course material is reinforced with a weekly assignment and quiz on the material covered. Both the weekly assignment and quiz were offered to students via the online course management system. A midterm and final exam was built from the material featured in the weekly assignments and quizzes. Grading was performed weekly, so students always knew where they stood in the course.

Office hours with the instructor were available both in person and via the online streaming platform of the student's choice. Students who were unable to meet during scheduled office hours are encouraged to email the instructor with the goal of an instructor response within twenty-four hours. Students who choose the asynchronous online learning model may feel disconnected from the campus environment and thus may particularly appreciate a quick response from their instructor.

**Learning Objectives**

Key learning objectives for the HyFlex version of CSCI 159: Computer Science Problem Solving were for students to learn how the field of Computer Science applies quantitative reasoning to analyze data, create algorithms, and solve challenging problems.

The course was divided into four modules. Students first learned the fundamentals of information systems and network infrastructure with assigned readings and facilitated discussion board reflections on their use and impact [21]. Learners defined and described information systems, the history of information systems, and information systems hardware. Each student was responsible for writing about the subcomponents of an information system, how it works, and how to troubleshoot failures within the system.

The next module introduced the information systems and technologies that make up the Internet. Students were introduced to web site development with HTML and were assigned a web development research project. For this, a website exploring a personal research topic was created demonstrating the use of at least ten different HTML tags.

The third module introduced algorithmic thinking, analysis, and how to read and write pseudocode. Students were introduced to algorithmic thinking and created algorithms based on problems they had previously solved [22]. Students then broke down their created algorithms into pseudocode, which prepared them for learning the fundamentals of programing with Python [23].

The last module focused on learning how to program in Python 3 using the freely available textbook "Think Python" [24]. Students were given weekly programming assignments so they could practice their skills with using variables, expressions, statements, and functions. Programing skills were paired with assignments focusing on problem solving and data analysis tasks. The PythonAnywhere environment provided a full Python development capacity in a web browser and allowed students to share their consoles with their instructor and other learners [16]. The Python turtle library provided excellent introduction in the use of functions and libraries [25]. Students designed and created Python programs to demonstrate algorithmic thinking and to solve creative problems that they chose.

All four modules featured weekly quizzes to assist students in reviewing selected knowledge and to have comprehension learning objectives integrated in the course.

In 1956, educational psychologist Benjamin Bloom created a classification of various levels of knowledge mastery that is known as "Bloom's Taxonomy" [26]. It included six levels for the cognitive domain. These are knowledge (information recall), comprehension (interpreting information), application (ability to use knowledge), analysis (breaking down knowledge), synthesis (bringing together knowledge components), and evaluation (make judgements) [27].

Table 1 maps Bloom's Taxonomy to the course's four modules. Modules with the learning objective of introducing knowledge used weekly quizzes that assessed students' ability to recall information. Modules that utilized discussion boards, classroom discussions, and reflective writing assignments that described or discussed concepts are mapped to the comprehension cognitive domain. Modules required students to utilize knowledge gained in the course to create within common computer science frameworks (such as algorithms and programming) and thus utilized the application cognitive domain. In the final weeks of the semester, when students paired their knowledge of algorithmic thinking with programming, bringing together parts of knowledge for new situations and problem solving, learners demonstrated the synthesis cognitive domain [27].

**Table 1. CSCI 159: Computer Science Problem Solving Learning Objectives mapped to Bloom's Taxonomy Cognitive Domains**

| Course Module | Learning Objective | Cognitive Domain Levels |
|---|---|---|
| Information Systems / Networks | Review knowledge, describe / discuss concepts | Knowledge, Comprehension |
| Internet & Web Development | Review knowledge, describe / discuss concepts, create HTML webpage | Knowledge, Comprehension, Application |
| Algorithmic Thinking | Review knowledge, describe / discuss concepts, create algorithms & pseudocode | Knowledge, Comprehension, Application |
| Programming with Python | Review knowledge, describe / discuss concepts, create programs | Knowledge, Comprehension, Application, Synthesis |

**Analysis of Learning Outcomes**

Based on observations and student feedback when CSCI 159: Computer Science Problem Solving was first offered using this model and an online format during the fall semester 2019 and throughout the coronavirus pandemic-impacted semesters of spring and fall 2020, it became clear that students placed a high value on course flexibility. This desire for flexibility was reflected in the design of the course. Many learners have work schedules that may not be compatible with instructor schedules or even their own desired course schedule. As the coronavirus pandemic started to impact student health, students could seamlessly switch modes and timeframes of course participation to keep up with the learning objectives. Weekly quizzes reinforced learning objectives and prepared students for scheduled midterm and final exams. Learners were able to use the weekly quizzes as exam study guides.

Course modules incorporated a mixture of weekly online discussion boards and assignments that had learners creating content based on the learning and course objectives. This resulted in a large volume of assignments needing grading and students seeking additional tutoring from the instructor and a graduate teaching assistant. In order to use a variety of open-ended, creative problem-solving assignments to reinforce the course learning objectives, the teaching assistant was an important student resource for larger classes.

Even when classrooms are once again fully safe for students and faculty, it will be valuable to integrate the content created for this approach to create a 'flipped classroom' synchronous learning version of this course.

## Conclusions

The HyFlex course delivery model allowed CSCI 159: Computer Science Problem Solving to be offered to a wide range of students. This approach offers the same course content, regardless if a student prefers the traditional classroom, synchronous, or asynchronous online learning approach. Using it, in future semesters, students can freely move between the physical classroom and online options, which is a significant benefit during a global pandemic in which students who would otherwise be in a classroom may need to attend online, by choice, or if asked to quarantine or if they become ill.

Online course content was specifically chosen to be platform-independent and perform well in a limited-bandwidth environment to enable student success. Course content was divided into four modules. Their learning objectives have been presented and evaluated in this paper.

Since supporting the HyFlex model required the course materials to be made available online, it is simple to adapt what was developed to use in the flipped classroom pedagogy, which has proven to be effective for teaching introductory courses in computer science and programming. Under all models, students value having course lectures available as reviewable recordings in the event they are ill, absent, or to enhance their studies with the course content. The approach to presenting CSCI 159 that was described herein is directly responsive to these needs.

## References


[1] D. Lederman, "One option for delivering instruction if campuses open this fall: HyFlex." https://www.insidehighered.com/digital-learning/article/2020/05/13/one-option-delivering-instruction-if-campuses-open-fall-hyflex (accessed Sep. 06, 2020).

[2] B. Beatty, "Beginnings - Hybrid-Flexible Course Design." https://edtechbooks.org/hyflex/book_intro (accessed Sep. 07, 2020).

[3] B. J. Beatty, "Hybrid-Flexible Course Design Costs and Benefits for Hybrid-Flexible Courses and Programs Is the value worth the effort associated with Hybrid-Flexible course

implementation? When is implementing a Hybrid-Flexible course worth the cost? The Value of a Student-Directed Hybrid," EdTech Books, 2019.

[4] "School closures caused by Coronavirus (Covid-19)." https://en.unesco.org/covid19/educationresponse (accessed Sep. 07, 2020).

[5] C. R. Kinlaw, L. L. Dunlap, and J. A. D'Angelo, "Relations between faculty use of online academic resources and student class attendance," *Computers and Education*, vol. 59, no. 2, pp. 167–172, Sep. 2012, doi: 10.1016/j.compedu.2011.12.028.

[6] E. J. Dommett, B. Gardner, and W. van Tilburg, "Staff and students perception of lecture capture," *Internet and Higher Education*, vol. 46, p. 100732, Jul. 2020, doi: 10.1016/j.iheduc.2020.100732.

[7] K. A. Babb and C. Ross, "The timing of online lecture slide availability and its effect on attendance, participation, and exam performance," *Computers and Education*, vol. 52, no. 4, pp. 868–881, May 2009, doi: 10.1016/j.compedu.2008.12.009.

[8] B. J. Beatty, "Hybrid-Flexible Course Design Values and Principles of Hybrid-Flexible Course Design," EdTech Books, 2019.

[9] "General Education < North Dakota State University." https://bulletin.ndsu.edu/past-bulletin-archive/2019-20/academic-policies/undergraduate-policies/general-education/#genedcoursestext (accessed Apr. 20, 2021).

[10] "Computer Science (CSCI) < North Dakota State University." https://bulletin.ndsu.edu/past-bulletin-archive/2019-20/course-catalog/descriptions/csci/ (accessed Apr. 20, 2021).

[11] B. Bernard, "CSCI 159: Computer Science Problem Solving Spring 2020 Syllabus," 2020.

[12] C. Stöhr, C. Demazière, and T. Adawi, "The polarizing effect of the online flipped classroom," *Computers and Education*, vol. 147, p. 103789, Apr. 2020, doi: 10.1016/j.compedu.2019.103789.

[13] A. Amresh, A. R. Carberry, and J. Femiani, "Evaluating the effectiveness of flipped classrooms for teaching CS1," in *Proceedings - Frontiers in Education Conference, FIE*, 2013, pp. 733–735, doi: 10.1109/FIE.2013.6684923.

[14] H. Jonsson, "Using flipped classroom, peer discussion, and just-in-time teaching to increase learning in a programming course," in *Proceedings - Frontiers in Education Conference, FIE*, Dec. 2015, vol. 2015, doi: 10.1109/FIE.2015.7344221.

[15] J. Elmaleh and V. Shankararaman, "Improving student learning in an introductory programming course using flipped classroom and competency framework," in *IEEE Global Engineering Education Conference, EDUCON*, Jun. 2017, pp. 49–55, doi: 10.1109/EDUCON.2017.7942823.

[16] “An online Python learning environment: PythonAnywhere.” https://www.pythonanywhere.com/details/education (accessed Sep. 07, 2020).

[17] “Table of Contents — How to Think like a Computer Scientist: Interactive Edition.” https://runestone.academy/runestone/books/published/thinkcspy/index.html (accessed Sep. 07, 2020).

[18] “Trinket: An Hour of Python.” https://hourofpython.com/ (accessed Sep. 07, 2020).

[19] “Open Textbooks | Student Government | NDSU.” https://www.ndsu.edu/sg/resources/oer/ (accessed Apr. 19, 2021).

[20] “Open Education Resources | NDSU Libraries.” https://library.ndsu.edu/guides/open-education-resources (accessed Apr. 20, 2021).

[21] D. Bourgeois, *Information Systems for Business and Beyond (2019)*. .

[22] “Algorithms to Live By - The Computer Science of Human Decisions by Brian Christian and Tom Griffiths.” https://algorithmstoliveby.com/excerpt.html (accessed Apr. 19, 2021).

[23] “Algorithmic Thinking: Changing How We Think About Human Improvement | by Dr. Jamie Schwandt | Medium.” https://jamieschwandt.medium.com/algorithmic-thinking-changing-how-we-think-about-human-improvement-670e362e7416 (accessed Apr. 19, 2021).

[24] “Think Python 2e – Green Tea Press.” https://greenteapress.com/wp/think-python-2e/ (accessed Apr. 19, 2021).

[25] “The Beginner’s Guide to Python Turtle – Real Python.” https://realpython.com/beginners-guide-python-turtle/ (accessed Apr. 19, 2021).

[26] C. Starr, B. Manaris, R. H. Stalvey, and C. W. Starr, “Bloom’s Taxonomy Revisited: Specifying Assessable Learning Objectives in Computer Science,” doi: 10.1145/1352135.1352227.

[27] “Learning objective - EduTech Wiki.” http://edutechwiki.unige.ch/en/Learning_objective (accessed Apr. 19, 2021).