THE GOVERNOR'S SCHOOL for SCIENCE AND TECHNOLOGY

New Horizons Governor's School for Science and Technology
Engineering, Design, Innovation, and Entrepreneurship (EDIE) Course Syllabus

Governor's School Course Information:

- CSC 221 — Introduction to Problem-Solving and Programming (Dual-enrolled)
- CSC 222 – Object-Oriented Programming (Dual-enrolled)
- CMDA 2014 – Data Matters (NOT dual-enrolled)

Instructor: Mr. Gaelan Venturi
Phone: 766-1100 x3392
Classroom & Office: C2B & C9
email: gaelan.venturi@nhrec.org

Communication with Instructor: In addition to time in class and the above contact information, the instructor can be reached via email (preferred) or CANVAS.

Course Description CSC 221: Introduces problem-solving and implementation of solutions using a high-level programming language in a structured programming environment. Includes concepts and practice of structured programming, problem-solving, top-down design of algorithms, a high-level programming language syntax, control structures, arrays, and an introduction into object-oriented programming. First course is in a three course sequence. (CSC 221-222-223) The assignments in this course require mathematical issue solving skills, algebraic modeling and functions, and use of variables. Lecture 3 hours. Total 3 hours per week. 3 credits

Course Description CSC 222: Introduces the concepts and techniques of object-oriented programming to students with a background in procedural programming and issue solving. Uses a high-level computer language to illustrate and implement the topics. Second course in a three — course sequence. (CSC 221-222-223). Lecture 4 hours. Total 4 hours per week. 4 credits

Course Description CMDA 2014: The intent for this course is to strike the balance of teaching modern, complex analytic methods to students who are almost entirely new to data analytics. For example, current statistics, computer science, CMDA, and other analytic courses that tend to focus on one data type when teaching analytic approaches. Data Matter, however, will teach one or two technical approaches for both collecting and summarizing data for different data types, e.g., quantitative data, text data, and image data. In doing so, we explicitly address analytic needs in the hard sciences, social sciences, and engineering sciences. Data Matter is intended to be available for all students, including those who find themselves challenged by STEM subjects.

Thus, the course does not have formal prerequisites, beyond high school math. This means that all skills needed to succeed in class will be taught during the class. However, the course might be easier for some students with some exposure to programming and learning from data.

General Course Purpose CSC 221 and 222: CSC 221 and 222 comprise the standard sequence of minimal programming content for computer science majors. The course sequence will teach the students to use high-level languages and their applications to problem-solving by using algorithms within procedural and object-oriented techniques, while ensuring data adheres to a structured model. This course is the first course in the sequence. It introduces computer-based problem-solving and implementation of solutions in a high-level programming language. The Introduction to Object-Oriented Programming course covers the topics of classes, objects, encapsulation, cohesion, inheritance, abstraction, and polymorphism. The course sequence will teach the students to use high-level languages and their applications to issue solve by using algorithms within procedural and object-oriented languages. C++/Python are the main languages used in this course.

Course Prerequisites/Corequisites CSC 221: None

Course Prerequisites/Corequisites CSC 222: Prerequisite – CSC 221
College Credit: Students may receive one semester of college credit (3 credit hours) for this course by dual enrolling at Virginia Peninsula Community College. Details will be provided during the first week of school.

While GSST will do all its power to secure dual enrollment (DE) status for its courses, dual-enrolled course credits are not guaranteed. Since the Virginia Community College System (VCCS) and Virginia Peninsula Community College set the criteria for DE and must approve each course and instructor, unavoidable circumstances that are not within the control of GSST may change the DE eligibility of any given GSST course. Alternative pathways for meeting specialty program requirements (E.g., concurrent associate's degree) should be discussed in advance with the home high school counselor.

Implications of Dual-Enrollment

Students must keep in mind that enrollment in a college class, including dual enrollment while in high school, entails consequences that can be significant and permanent, including, but not limited to, the following:

- Final course grades on college transcripts become a permanent part of a student's college record.
- Graduate-level education programs may consider DE grades equally with traditional college courses in calculating GPA for admission (E.g., graduate, medical, veterinary schools).
- Grades of D and F and course withdrawals can negatively affect scholarship and financial aid requests.
- Once the withdrawal date has passed, students cannot withdraw from a class, except in extraordinary circumstances such as a medical emergency.

Textbook: No textbook required for this course
Materials: The course is designed around the use of computers in the classroom

Course Meetings:

- AM Governor's School:
    - MTRF, 8:50 AM – 10:25 AM; W 8:40 AM – 10:25 AM
- PM Governor's School:
    - MTRF, 11:20 AM — 12:55 PM; W 11:20 PM – 1:05 PM

Grading Scale:

- A = 90-100%
- B = 80-89%
- C = 70-79%,
- D = 60-69%
- F = less than 60%

Quarterly Grades:

- Projects = 10%
- Classwork = 25%
- Quizzes = 25%
- Tests = 40%

Semester Grade Determination:

- S1 Grade = 50% Q1 + 50% Q2
- S2 Grade = 50% Q3 + 50% Q4
- Yearlong Grade = 50%S1 + 50%S2

Late Assignments: Assignments will be submitted via Canvas LMS, except for some assignments and quizzes/tests that will be submitted on paper. Assignments should be submitted on or in advance of the due date and time.

*If an assignment is submitted late without an excused absence, the grade will have a resulting penalty of **25%** per day that it is late.*

Absences: Missing class is strongly discouraged! A great deal of learning occurs during each class period, whether we are meeting face-to-face or virtually. In addition, the impact is not just on the absent student, but on laboratory partners and on the class, who are deprived of the missing student's contributions to discussions.

- If an absence is unexpected (i.e., due to illness, family emergency, etc.) parent/guardian communication is expected within a day of the absence to explain the absence. Subsequent communication between the instructor and student will facilitate an agreement on the terms of make-up work.
- In the event of a long-term absence, alternate assignments will be made to accommodate student needs. Supporting optimal learning potential, the instructor will determine a reasonable timeframe for make-up work to be submitted to gain credit for the submissions.

Academic Integrity Guidelines:

All students are required to take responsibility for upholding everyone's honesty in the classroom. All students will sign a copy of the GSST Honor Pledge during the first week of school.

The Pledge: *"I pledge to support the Governor's School for Science and Technology (GSST) Code for Academic Work. I will refrain from any dishonesty or deception, such as cheating or plagiarizing, which are honor code violations, on all academic work. I am further aware that as a member of the academic community, I should report any suspected violations to an instructor."*

No form of cheating, copying, or plagiarizing will be tolerated. Homework and class assignments are independent work and should not be copied from any source, including AI sources, such as ChatGPT. Students may share lab data collected by members of their group; however, lab reports and all other written assignments are to be done independently unless the assignment clearly states that it is group work. The level of collaboration allowed between students will be indicated on each assignment. In addition, students will read and sign the Permissible Assistance form to clarify types of assistance that are encouraged vs. not tolerated in this course.

In the case of copying, there will be no determination of who copied from whom; all students involved will receive no credit for the assignment and the students involved may be referred to the GSST administration for disciplinary action. **Detection of AI-generated responses will result in no credit for the assignment, and a parent conference will be scheduled. Use of AI for assignments is conditional on the instructor's discretion.** Regarding tests and quizzes, if students share information during an assessment or look at notes, internet sources, or other materials during the assessment, all students involved will receive no credit for the assessment, and the students will be referred to the school's administration for disciplinary action.

To avoid plagiarism, all research sources must be cited properly to give the author(s) credit. In addition, such information will be summarized or paraphrased, never just copied from its source.

The goal of the Governor's School is not only to help students gain acceptance to top colleges and access learning and career opportunities, but to thrive and excel once they have gained that entry. To this end, we take the academic integrity of each of our students very seriously.

Guidelines for Permissible Assistance on Assignments in Computational Science

Students: If you have any questions now or during the year about acceptable assistance, ask your teacher.

Type of Assignment	What Type of Assistance is permitted? (Types of assistance include, but are not limited to, classmates, parents, mentors, books, and the Internet.)
Assignments	Collaboration between classmates is encouraged, but unless you are in a group that is only submitting one copy of an assignment, or you are otherwise instructed, each assignment submitted must be your own work.
Quizzes	The quizzes given are expected to be completed on your own. No outside assistance is allowed, including, but not limited to, family, friends, the internet, past homework/classwork, past notes, Morse code taps, AI, etc.
Projects	Projects that are not group-based require you to submit your own work. Group projects where only 1 copy of the submission is required are allowed to have the same work; otherwise, your own work must be submitted.
Tests	Tests are completed individually in class with no outside assistance. All work submitted is yours and yours alone. No outside assistance is allowed, including, but not limited to, family, friends, the internet, past homework/classwork, past notes, Morse code taps, AI, etc.

Student Learning Outcome 1: Civic Engagement

- Engage and build technology that responds to human needs and helps people navigate institutional systems.

Student Learning Outcome 2: Critical Thinking

- Assess why certain solutions might not work and to save time in coming up with a more efficient approach.

Student Learning Outcome 3: Professional Readiness

- Work well with others and display situationally and culturally appropriate demeanor and behavior.

Student Learning Outcome 4: Quantitative Literacy

- Perform accurate calculations, interpret quantitative information, apply and analyze relevant numerical data, and use results to support conclusions.

Student Learning Outcome 5: Scientific Literacy

- Represent real-world objects and processes virtually by identifying properties, behavior, and operations relevant to solving problems on a computer.

Student Learning Outcome 6: Written Communication

- Develop, convey, and exchange ideas in writing, as appropriate to a given context and audience.

Student Learning Outcome 7: Basic Concepts of Computer Systems

- Differentiate computer components by functionality and define x.

Student Learning Outcome 8: Review of Procedural Problem-Solving Concepts

- Describe activities related to program development
- Solve problems using techniques such as pseudocode, flowcharts, UML, and model development.
- Evaluate algorithms for errors
- Discuss the presence of algorithms in various activities

Student Learning Outcomes 9: Review of Procedural Programming

- Describe activities related to program development
- Solve problems using techniques such as pseudocode, flowcharts, UML, and model development.
- Evaluate algorithms for errors
- Discuss the presence of algorithms in various activities

Student Learning Outcomes 10: Object-Oriented Design

- Design programs using appropriate program design techniques.
- Develop programs using sequential and selection operations
- Choose adequate repetition structures based on the type of application
- Solve problems using procedures
- Develop applications using arrays.

Student Learning Outcomes 11: Development and Testing Tools

- Apply various tools for program development and testing.
- Apply a version control system in team or multiple revision scenarios.
- Apply the use of an automated debugger to set breakpoints and examine data values.

Student Learning Outcomes 12: Abstract data type (ADT) Implementations and Applications

- Design and implement classes
- Design, implement, and manipulate objects belonging to classes
- Explain the difference between data structures that are internal versus external to a class.

Student Learning Outcomes 13: Recursion

- Explain the parallels between ideas of mathematical and structural induction to recursion and recursively defined structures.
- Create a simple program that uses recursion.
- Describe how recursion is implemented on a computer.

Student Learning Outcomes 14: Inheritance & Polymorphism

- Explain the benefits and restrictions of inheritance
- Distinguish between inheritance of implementation and inheritance of design
- Design class hierarchies using inheritance and interfaces.
- Create a class which implements an interface.

Student Learning Outcomes 15: Files & Exceptions

- Create programs using file handling techniques
- Describe the use of relative and absolute paths to identify a file.
- Detecting end of input conditions and common error conditions.
- Explain encapsulating exceptions
- Demonstrate throwing and catching exceptions
- Write code to implement try catch and finally blocks
- Write code to create a custom Exception.

Student Learning Outcome 16: Review of Object-Oriented Principles

- Compare and contrast procedural versus object-oriented programming
- Design class hierarchies using inheritance and interfaces
- Implement in code OOP constructs including encapsulation, inheritance, and polymorphism
- Review the design, implementation, and efficiency of recursive algorithms
- Review of arrays and exception handling

Student Learning Outcomes 17: Analysis of Algorithms

- Discuss the differences between iterative vs. recursive algorithms
- Demonstrate worst-case complexity function
- Define other complexity functions such as best case, average case, and amortized.

Student Learning Outcomes 21: CMDA 2014 – Data Matters

- Recite an analytic pipeline.
- Analyze data of varying types and from varying disciplines.
- Formulate questions/hypotheses.
- Identify and challenge biases that appear at all phases of data analyzes, including data collection,
- choice in analytical methods, implications.
- Recognize ethical issues and vulnerabilities when learning from data and extrapolating findings to
- larger populations.

- Communicate findings from data effectively, responsibly, and ethically.
- Collect relevant, high-dimensional data and information efficiently and ethically and from multiple
- sources.
- Program in R.

Major Topics to be included:

- Basic concepts of computer systems
- Processing Code
- Problem analysis and algorithmic modeling
- Use of data
- Decision structures
- Repetition structures
- Programming with Procedures
- Classes and Introduction to Libraries
- Arrays
- Review of Procedural Problem-Solving Concept
- Object-Oriented Design
- Development & Testing Tools
- Abstract data type (ADT) Implementations & Applications
- Recursion
- Inheritance & Polymorphism
- Files & Exceptions
- Review of Object-Oriented Principles
- Analysis of Algorithms
- Data Structures
- Searching and Sorting Algorithms
- Additional Data Structures
- Real-World Applications

Please return this page and this page only to your teacher after you and a parent/guardian have read the syllabus, the GSST Honor Pledge, and the Permitted Assistance form by September 2nd, 2025.

I have read the syllabus for Engineering, Design, Innovation, and Entrepreneurship at GSST. I will contact Mr. Venturi by phone at (757)-766-1100, x3392, via CANVAS, or by email gaelan.venturi@nhrec.org if I have any questions.

Student’s Printed Name: ______________________________

Student’s Signature: _________________________________

Parent/Guardian’s Signature: ___________________________