FREEHOLD REGIONAL HIGH SCHOOL DISTRICT

OFFICE OF CURRICULUM AND INSTRUCTION

MATHEMATICS DEPARTMENT

HONORS COMPUTER SCIENCE

Grade Level: 10-12

Credits: 2.5

BOARD OF EDUCATION ADOPTION DATE:

AUGUST 31, 2015

FREEHOLD REGIONAL HIGH SCHOOL DISTRICT

Board of Education

Mr. Heshy Moses, President
Mrs. Jennifer Sutera, Vice President
Mr. Vincent Accettola
Mr. William Bruno
Mrs. Elizabeth Canario
Mr. Samuel Carollo
Mrs. Amy Fankhauser
Mrs. Kathie Lavin
Mr. Michael Messinger

Central Administration

Mr. Charles Sampson, Superintendent
Dr. Nicole Hazel, Chief Academic Officer
Dr. Jeffrey Moore, Director of Curriculum and Instruction
Ms. Stephanie Mechmann, Administrative Supervisor of Curriculum & Instruction
Dr. Nicole Santora, Administrative Supervisor of Curriculum & Instruction

Curriculum Writing Committee

Mr. James Gill
Ms. Rhonda Solomon

Supervisors

Ms. Deanna Farinick
Ms. Angelique Gauthier
Mr. Joseph Iacullo
Ms. Annette Kinsley
Ms. Jennifer Okerson
Ms. Denise Scanga

HONORS COMPUTER SCIENCE
COURSE PHILOSOPHY
In *Honors Computer Science*, students will enhance their critical thinking and problem-solving skills and learn the fundamentals of computer programming in a fast-paced and rigorous environment. Through the creation of user-friendly programs that model and solve real-world scenarios and problems, students will develop skills in computer programming that are the foundation for understanding all programming languages. A focus of this course is to challenge the novice programmer to create effective, robust computer programs.
COURSE DESCRIPTION
Honors Computer Science is an introductory computer science course. It provides the foundational knowledge and skills that students will need to be successful in AP Computer Science A. Students will learn the basic structures of programming such as repetition statements, selection statements, and arrays. They will learn these basics through the creation of user-friendly and efficient programs using the Java programming language.

COURSE SUMMARY	
COURSE GOALS	
CG1: Students will design solutions to complex problems by developing algorithms. CG2: Students will apply their learning of programming concepts to decipher other programming languages. CG3: Students will create programs that adhere to complex specifications. CG4: Students will effectively design solutions to complex tasks by modularizing them into smaller ones.	
COURSE ENDURING UNDERSTANDINGS	COURSE ESSENTIAL QUESTIONS
CEU1: There are multiple avenues to design solutions to problems and some may be more efficient than others.	CEQ1a: Can there be more than one algorithm to solve a problem and how can you determine whether one is better than another? CEQ1b: Can a program be effective while not efficient? CEQ1c: What is the meaning of “robust” and “elegant” in the world of programming?
CEU2: There are similarities and differences between all computer programming languages.	CEQ2: What are the essential similarities between programming languages? How does this help us to learn a language? CEQ2b: What are the similarities between programming and spoken languages?
CEU3: Writing a robust computer program requires the consideration of the specifications as well as predicting for unforeseen events.	CEQ3: What determines the use of decision making in program design?
CEU4: Writing efficient programs require that a problem be separated into independent modules.	CEQ4a: How do you go about solving a problem? CEQ4b: What makes a program maintainable?

UNIT GOALS & PACING		
UNIT TITLE	**UNIT GOALS**	**RECOMMENDED DURATION**
Unit 1: Introduction to Computer Programming	Students will effectively and efficiently create interactive programs by using algorithms and manipulating data with the use of variables.	4-6 weeks
Unit 2: Selection and Repetition Statements	Students will effectively and efficiently create interactive programs by incorporating selection and repetition statements.	5-7 weeks
Unit 3: Arrays, Methods, and Sorting	Students will effectively and efficiently incorporate arrays and write methods that will enhance the development of sorting and interactive programs.	5-7 weeks

HONORS COMPUTER SCIENCE
UNIT 1: Introduction to Computer Programming **SUGGESTED DURATION: 4-6 weeks**

UNIT OVERVIEW	
UNIT LEARNING GOALS	
Students will effectively and efficiently create interactive programs by using algorithms and manipulating data with the use of variables.	
UNIT LEARNING SCALE	
4	In addition to score 3 performances, the student can develop programs from more complex algorithms.
3	The student can: • describe the relationship between hardware and software; • describe the steps involved in program compilation and execution; • write an algorithm that is an ordered sequence of instructions for solving a problem; • write an algorithm when presented with a problem; • define the difference between primitive data and objects; • declare and use variables; • write programs using operations involving primitive data and the use of an input class.
2	The student sometimes needs assistance from a teacher, makes minor mistakes, and/or can do the majority of level 3 performances.
1	The student needs assistance in attempting to reach level 3.
0	Even with help, the student does not exhibit understanding of performances listed in level 3.

ENDURING UNDERSTANDINGS	ESSENTIAL QUESTIONS
EU1: An algorithm helps programmers formulate a plan to consider specifications as well as predict for unforeseen events.	EQ1: Could you efficiently write a program without first creating an algorithm?
EU2: A computer executes instructions exactly as written, not necessarily as intended.	EQ2: How do you confirm that what is written is what was intended by the programmer?
EU3: Determining data types when programming is integral to a program's effectiveness.	EQ3: How do you decide on what data type to use?
EU4: Programs that interact with users must consider the various possible human responses.	EQ4a: What if the user enters incorrect input? EQ4b: Why use live data?

NJCCCS & COMMON CORE STANDARDS

HSN-Q.A.2 Define appropriate quantities for the purpose of descriptive modeling.

SMP1 Make sense of problems and persevere in solving them.
SMP4 Model with mathematics.
SMP6 Attend to precision.
SMP7 Look for and make of structure.
SMP8 Look for and express regularity in repeated reasoning.

COMMON ASSESSMENT	
ALIGNMENT	DESCRIPTION
LG 1 EU1,EQ1 EU2,EQ2 EU3, EQ3 EU4,EQ4a,EQ4b HSN-Q.A.2 SMP 1, 4, 6, 7, 8 DOK 2	Quadratic Formula: Students will create a program that finds the roots of a quadratic equation when provided with the coefficients. Before creating the program, the students will have to create the algorithm for the program. Students must also review and be able to prove that their algorithm and their program take into consideration all possible coefficients.

SUGGESTED STRATEGIES		
ACTIVITIES	DECLARATIVE KNOWLEDGE	PROCEDURAL KNOWLEDGE
Students will hand write an algorithm of their choice (i.e., how to tie a tie). Student will select another student to actually follow the instructions written to determine whether the algorithm was written correctly.  The level of expectation for individual student's algorithm will be based on their ability.	parts of a computer system algorithm	Create an algorithm from a problem set DOK 3
Students will be directed to write code utilizing print statements that will generate various output.  The level of expectation for individual student's program will be based on their ability.	print statements variables	Develop a program from a simple algorithm and output certain information DOK 2

HONORS COMPUTER SCIENCE

UNIT 2: Selection and Repetition Statements **SUGGESTED DURATION: 5-7 weeks**

UNIT OVERVIEW	
UNIT LEARNING GOALS	
Students will effectively and efficiently create interactive programs by incorporating selection and repetition statements.	
UNIT LEARNING SCALE	
4	In addition to score 3 performances, the student can: • use more advanced conditional statements; • optimize nested if-statements to conjunctive Boolean statements resulting in more concise code.
3	The student can: • effectively use if/else statements; • use Boolean expressions when writing conditional statements; • appropriately incorporate loops in program design; • effectively use while statements; • effectively use for statements; • alter the flow of control of a program through the use of selection statements.
2	The student sometimes needs assistance from a teacher, makes minor mistakes, and/or can do the majority of level 3 performances.
1	The student needs assistance in attempting to reach level 3.
0	Even with help, the student does not exhibit understanding of performances listed in level 3.

ENDURING UNDERSTANDINGS	ESSENTIAL QUESTIONS
EU1: Program development can be enhanced by altering the flow of control via the use of conditional statements.	EQ1a: How do I choose which selection structure should be used to best facilitate the program? EQ1b: How do I determine when it is appropriate to execute statements out of sequence?
EU2: There are multiple ways to design and implement repeated operations.	EQ2: Is one method of looping more efficient or effective than another?

NJCCCS & COMMON CORE STANDARDS
HSN-Q.A.2 Define appropriate quantities for the purpose of descriptive modeling. SMP1 Make sense of problems and persevere in solving them. SMP4 Model with mathematics. SMP6 Attend to precision. SMP7 Look for and make of structure. SMP8 Look for and express regularity in repeated reasoning.

COMMON ASSESSMENT	
ALIGNMENT	DESCRIPTION
LG 1 EU1,EQ1a,EQ1b EU2,EQ2 EU3, EQ3 EU4,EQ4a,EQ4b HSN-Q.A.2 SMP 1, 4, 6, 7, 8 DOK 3	Sum It Up: Students will design and implement an application that reads an integer value and prints the sum of all the even integers between 2 and the input value, inclusive. The program will print an error message if the input value is less than 2. It will also prompt the user accordingly. Students will have to justify which type of loop they chose and why.

SUGGESTED STRATEGIES		
ACTIVITIES	DECLARATIVE KNOWLEDGE	PROCEDURAL KNOWLEDGE
Students will complete truth tables applying the principles of Boolean logic.	Boolean values logical operators	Create a truth table and evaluate expressions using logical operators DOK 3
Students will be asked to create a program that deliberately includes an endless loop.  The level of expectation for individual student's program will be based on their ability.	for loop while loop	Create programs using different kinds of loops DOK 3

HONORS COMPUTER SCIENCE

UNIT 3: Arrays, Methods, and Sorting **SUGGESTED DURATION: 5-7 weeks**

UNIT OVERVIEW

UNIT LEARNING GOALS

Students will effectively and efficiently incorporate arrays and write methods that will enhance the development of sorting and interactive programs.

UNIT LEARNING SCALE

4	In addition to score 3 performances, the student can use the index of an array to organize and store data.
3	The student can: • write programs that use arrays to represent many data elements; • write programs and methods to modularize larger, more complex code; • write programs to sort data using iterative sorting methods including insertion and selection sorts; • write programs to perform a sequential search; • write programs to perform a binary search on a sorted array; • use parallel arrays, using one array to maintain the status of the second array.
2	The student sometimes needs assistance from a teacher, makes minor mistakes, and/or can do the majority of level 3 performances.
1	The student needs assistance in attempting to reach level 3.
0	Even with help, the student does not exhibit understanding of performances listed in level 3.

ENDURING UNDERSTANDINGS	ESSENTIAL QUESTIONS
EU1: An array is a simple but powerful way to group and organize data that holds a fixed number of values of the same type.	EQ1: What requirements would insinuate that arrays would be beneficial in a program?
EU2: A method is an optimal way to compartmentalize parts of a program.	EQ2: What are the advantages of compartmentalization?
EU3: There are many different methods of sorting; some may be more effective or efficient than others depending on the data.	EQ3: What criteria would you use to determine which sort is the most effective?

NJCCCS & COMMON CORE STANDARDS

HSN-Q.A.2 Define appropriate quantities for the purpose of descriptive modeling.
HSN.VM.C.6 Use matrices to represent and manipulate data.

SMP1 Make sense of problems and persevere in solving them.
SMP4 Model with mathematics.
SMP6 Attend to precision.
SMP7 Look for and make of structure.
SMP8 Look for and express regularity in repeated reasoning.

COMMON ASSESSMENT	
ALIGNMENT	DESCRIPTION
LG 1 EU1,EQ1 EU2,EQ2 EU3, EQ3 HSN-Q.A.2 HSN.VM.C.6 SMP 1, 4, 6, 7, 8 DOK 3	Student Grade Retrieval: Students will create a program that uses parallel arrays to maintain student names and corresponding grades in a specific course. The program will search for a name and print out the name and corresponding grade. The students will need to be able to explain why they must use a sequential search as opposed to a binary search to successfully code this program
LG 1 EU1,EQ1 EU2,EQ2 EU3, EQ3 HSN-Q.A.2 HSN.VM.C.6 SMP 1, 4, 6, 7, 8 DOK 3	Teachers' EZ Grader: Students will write a program that asks the user to enter in 10 grades. The grades will be stored in an array. The array of grades will be sent to a method that will sort the array in descending order. A second method will be called that calculates the average grade. The driver program will be responsible to print out the sorted array and the average grade. Students will have to justify which type of sort they used and why.

SUGGESTED STRATEGIES		
ACTIVITIES	DECLARATIVE KNOWLEDGE	PROCEDURAL KNOWLEDGE
Students will participate in a human sort. They will stand in line in random height order, and by using the selection sort algorithm they will sort themselves from shortest to tallest.	selection sort	Use the selection sort DOK 3
Students will code and execute a selection sort and determine the number of comparisons made.  The level of expectation for individual student's program will be based on their ability.	selection sort	Create a program that involves selection sort DOK 3
Students will play a game with a partner. The partner will choose a number from 1-100. Using the binary search, the student must guess the number in the smallest amount of guesses possible	binary search	Use the binary search DOK 3