COURSE TITLE

Computer Programming C++

LENGTH

One Semester
Grades 10-12

DEPARTMENT

Computer Department
Barbara O'Donnell, Supervisor

SCHOOL

Rutherford High School

DATE

September 10, 2018

Initial BOE Approval Date (Born on): 5/11/2015

COMPUTER PROGRAMMING C++

I. Introduction/Overview/Philosophy

C++ is one of the most widely used programming languages for developing portable application software. This course introduces students to problem-solving and program development using object-oriented design, structured programming techniques, and the C++ programming language. Specific topics include data types, control statements, functions, argument passing, arrays, and strings. Students will design, construct, and test programs with primarily scientific, mathematical, and business applications.

II. Objectives

Course Outline:

I. Preview Computer Science
- A. Understand that computer science is interdisciplinary
- B. Discuss the historical development of computers
- C. Know what constitutes computer hardware/software
- D. Observe differences between types of computer languages
- E. Discuss the evolution of C++
- F. Computer and Information Ethics
- G. Compiling a simple program
- H. The compilation process
- I. Algorithms

II. Numbers and Objects
- A. Vocabulary of C++
  1. Reserved words
  2. Library identifiers
  3. Basic program components
- B. Number types
  1. Numeric range and precision
  2. Variable definition
  3. Output statement
- C. Input
  1. Format
  2. Prompt
  3. Buffered / Failed
- D. Assignment
  1. Expression
  2. Right into left
  3. Roundoff errors
  4. Types / Casts
  5. Combining assignment and arithmetic
  6. Incrementing / Decrementing
- E. Constants
  1. Definition / Format
  2. Proper use
  3. Enumerated types
- F. Arithmetic
  1. Symbols / Meanings
  2. Order of operations

3. Modulus
4. Integer division
5. Unbalanced parentheses
6. Mathematical functions

G. Strings
1. String variables
2. String functions
3. Concatenation
4. Characters and strings
5. Formatted output

III. Control Flow

A. The If statement
1. Purpose
2. Block statement
3. Brace layout

B. Relational operators
1. Notation / Description
2. = vs ==
3. Compile errors
4. Comparison of floating point numbers
5. ASCII and Comparison of strings

C. Multiple alternatives
1. Two-way branch / branches
2. Order of Conditions
3. The switch statement
4. Dangling else

D. Nested branches
1. Two-level decision process
2. Multi-level decision process

E. Boolean operations
1. && / || / !
2. Left to right evaluations / Short-circuiting
3. Multiple relational operators
4. Combining && and || Conditions
5. DeMorgan's Law

F. The "while" loop
1. Format
2. Infinite loops
3. Off-by-one error

G. The "for" loop
1. Format
2. Symmetric and asymmetric bounds
3. Count iterations
4. Incorrect use of !=

H. The "do" loop
1. Purpose
2. Initialization of values

I. Nested loops
1. Patterns
2. Processing Inputs

IV. Subprograms: functions for problem solving
- A. Purpose
  - 1. Rationale for program design
  - 2. Use of functions
- B. User-defined functions
  - 1. Built-in vs user-defined
  - 2. Function declarations
- C. Implementing functions
  - 1. Name and type
  - 2. Call to function
  - 3. Reuse
  - 4. Side effects
- D. Function comments
  - 1. Relevance
  - 2. Pre-condition / Post-condition
  - 3. Identify parameters
  - 4. Return values
- E. Return values
  - 1. Termination of function
  - 2. Missing return value
- F. Parameters
  - 1. Formal / Actual
  - 2. Call by value and call by reference
  - 3. Meaningful Names
  - 4. Type Mismatch
- E. Scope of identifiers
  - 1. Local / global variables

V. Arrays
- A. Basic concept and notation for arrays
  - 1. Definition and usage
- B. Input/output of arrays
- C. Arrays in functions
  - 1. Array parameters
- D. Searching an array
- E. Selection sort

Student Outcomes:

Upon completion of the course, students will demonstrate the ability to:

- describe basic computer organization.
- identify various types of computer programming languages.
- demonstrate problem-solving skills necessary for algorithm development.
- incorporate modular and object-oriented design into problem-solving tasks.
- develop C++ programs using structured and object-oriented programming techniques.
- construct programs using the C++ programming language syntax.
- test C++ programs to verify correctness.
- formulate, manipulate, and evaluate arithmetic expressions.
- identify the C++ language data types.
- declare C++ variables.

- manipulate C++ variables.
- construct C++ output statements.
- incorporate modular and object-oriented design through the utilization of functions and methods.
- recognize selection structures.
- design, construct, and implement selection structures in C++ programs.
- recognize looping structures.
- design, construct, and implement looping structures in C++ programs.
- declare, construct, and process one-and two-dimensional arrays.
- declare, initialize, and manipulate pointer variables.
- recognize the use of pointers for call-by-reference functions.
- utilize string processing techniques and functions.
- describe object-oriented program abstractions and design.
- analyze and implement C++ classes.
- implement user-designed functions.
- examine computer and information ethics
- explore history of the computer.

New Jersey Student Learning Standards

CAREER READY PRACTICES

CRP1 Act as a responsible and contributing citizen and employee

Career-ready individuals understand the obligations and responsibilities of being a member of a community, and they demonstrate this understanding every day through their interactions with others. They are conscientious of the impacts of their decisions on others and the environment around them. They think about the near-term and long-term consequences of their actions and seek to act in ways that contribute to the betterment of their teams, families, community and workplace. They are reliable and consistent in going beyond the minimum expectation and in participating in activities that serve the greater good.

CRP2 Apply appropriate academic and technical skills

Career-ready individuals readily access and use the knowledge and skills acquired through experience and education to be more productive. They make connections between abstract concepts with real-world applications, and they make correct insights about when it is appropriate to apply the use of an academic skill in a workplace situation.

CRP4 Communicate clearly and effectively and with reason.

Career-ready individuals communicate thoughts, ideas, and action plans with clarity, whether using written, verbal, and/or visual methods. They communicate in the workplace with clarity and purpose to make maximum use of their own and others' time. They are excellent writers; they master conventions, word choice, and organization, and use effective tone and presentation skills to articulate ideas. They are skilled at interacting with others; they are active listeners and speak clearly and with purpose. Career-ready individuals think about the audience for their communication and prepare accordingly to ensure the desired outcome.

CRP6 Demonstrate creativity and innovation

Career-ready individuals regularly think of ideas that solve problems in new and different ways, and they contribute those ideas in a useful and productive manner to improve their organization. They can consider unconventional ideas and suggestions as solutions to issues, tasks or problems, and they discern which ideas and suggestions will add greatest value. They seek new methods, practices, and ideas from a variety of sources and seek to apply those ideas to their own workplace. They take action on their ideas and understand how to bring innovation to an organization.

CRP8 Utilize critical thinking to make sense of problems and persevere in solving them
Career-ready individuals readily recognize problems in the workplace, understand the nature of the problem, and devise effective plans to solve the problem. They are aware of problems when they occur and take action quickly to address the problem; they thoughtfully investigate the root cause of the problem prior to introducing solutions. They carefully consider the options to solve the problem. Once a solution is agreed upon, they follow through to ensure the problem is solved, whether through their own actions or the actions of others.

CRP11 Use technology to enhance productivity
Career-ready individuals find and maximize the productive value of existing and new technology to accomplish workplace tasks and solve workplace problems. They are flexible and adaptive in acquiring new technology. They are proficient with ubiquitous technology applications. They understand the inherent risks-personal and organizational-of technology applications, and they take actions to prevent or mitigate these risks.

TECHNOLOGY STANDARDS
STANDARD 8.1: EDUCATIONAL TECHNOLOGY: ALL STUDENTS WILL USE DIGITAL TOOLS TO ACCESS, MANAGE, EVALUATE, AND SYNTHESIZE INFORMATION IN ORDER TO SOLVE PROBLEMS INDIVIDUALLY AND COLLABORATE AND TO CREATE AND COMMUNICATE KNOWLEDGE.
8.1.12.B.2 - Apply previous content knowledge by creating and piloting a digital learning game or tutorial.
8.1.12.D.5 - Analyze the capabilities and limitations of current and emerging technology resources and assess their potential to address personal, social, lifelong learning, and career needs.
8.1.12.F.1 - Evaluate the strengths and limitations of emerging technologies and their impact on educational, career, personal and or social needs.

STANDARD 8.2: TECHNOLOGY EDUCATION, ENGINEERING, DESIGN, AND COMPUTATIONAL THINKING – PROGRAMMING: ALL STUDENTS WILL DEVELOP AN UNDERSTANDING OF THE NATURE AND IMPACT OF TECHNOLOGY, ENGINEERING, TECHNOLOGICAL DESIGN, COMPUTATIONAL THINKING, AND THE DESIGNED WORLD AS THEY RELATE TO THE INDIVIDUAL, GLOBAL SOCIETY, AND THE ENVIRONMENT.
8.2.12.E.3 - Use a programming language to solve problems or accomplish a task (e.g., robotic functions, website designs, applications, and games).
8.2.12.E.4 - Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstraction, variables, data types and conditional statements).

STANDARD 9.3: CAREER AND TECHNICAL EDUCATION
PATHWAY: PROGRAMMING & SOFTWARE DEVELOPMENT (IT-PRG)
9.3.IT-PRG.6 Program a computer application using the appropriate programming language.

III. Proficiency Levels

This course is open to grades 10-12. A passing grade in Computer Programming 1 is required to take this course.

IV. Methods of Assessment

Student Assessment

The teacher will provide a variety of assessments during the course of the year. Among these are homework, laboratory exercises, teacher made tests and quizzes, and long-term projects.

Curriculum/Teacher Assessment

The teacher will provide the subject area supervisor with suggestions for changes on an ongoing basis.

V. Grouping

The prerequisite for Computer Programming C++ is successful completion of Computer Programming 1.

VI. Articulation/Scope & Sequence/Time Frame

Course length is one semester and is offered to students in grades 10-12.

VII. Resources

Adams, Joel, Leestma, Sanford, and Nyhoff, Larry C++ An Introduction to Computing Upper Saddle River, NJ: Prentice Hall, 1995.

Astrachan, Owen L. A Computer Science Tapestry New York: McGraw Hill Inc., 1996.

Bergin, Joseph, Stehlik, Mark, Roberts, Jim, and Pattis, Richard C++ A Gentle Introduction to the Art of Object-Oriented Programming New York: John Wiley and Sons, Inc., 1997.

Dale, Nell A Laboratory Course in C++ Sudbury, MA. Jones and Bartlett Publishers Inc., 1997.

Horstmann, Cay, Budd, Timothy, Big C++ 2nd Edition: Wiley & Sons Inc., 2009.

Lambert, Kenneth and Nance, Douglas Understanding Programing and Problem Solving with C++. Boston: PWS Publishing Company, 1997.

VIII. Suggested Activities

- Laboratory programming problems
- Class presentations
- Cooperative programming projects

IX. Methodologies

Much of the class time is spent in lab work and on programming problems to be completed. When group instruction is necessary, topics are taught using the computer projection system, in conjunction with student classwork.

X. Interdisciplinary Connections

Connections are made to mathematics by using a variety of arithmetic formulas, as well as higher mathematical concepts. Connections are also made to the disciplines of art and English, by means of incorporation of theses ideas into programming projects.

XI. Differentiating Instruction for Students with Special Needs: Students with Disabilities, Students at Risk, English Language Learners, and Gifted & Talented Students

Differentiating instruction is a flexible process that includes the planning and design of instruction, how that instruction is delivered, and how student progress is measured. Teachers recognize that students can learn in multiple ways as they celebrate students' prior knowledge. By providing appropriately challenging learning, teachers can maximize success for all students.

Differentiating in this course includes but is not limited to:

Differentiation for Support (ELL, Special Education, Students at Risk)

- activity choice
- appeal to diverse learning styles
- choice to work with others or alone
- hands-on activities
- multimodal activities
- advance organizers
- pre-teaching vocabulary
- visual demonstrations, illustrations, and models
- work with checklists
- online video review
- peer teaching and support
- study guides
- powerpoint reference
- partnering
- guided notetaking
- reteach and review
- peer mentoring on lab
- demonstrations on smartboard

Differentiation for Enrichment

- more complex tasks and problems
- higher expectations in assessment questioning
- independent extensions based on student interest, curiosity, and choice
- extended research and readings
- curriculum compacting
- flexible grouping on challenging exercises
- higher level questioning techniques
- facilitating classmates
- higher expectation for writing programs
- display and explain version of coded task
- peer mentoring

XII. Professional Development

The teacher will continue to improve expertise through participation in a variety of professional development opportunities.

XIII. Curriculum Map/Pacing Guide

The course utilizes a blended classroom approach offered through CodeHS. The content is fully web-based.

Unit Topic	Time Allocated	Differentiating Instruction for Students with Disabilities, Students at Risk of School Failure, English Language Learners, & Gifted & Talented Students	Standards	Assessments
Introduction to Computer Science/ Programming in C++ and Digital Citizenship • History of Computers • Computer and Information Ethics • Computer science is interdisciplinary • Historical development of computers • Computer hardware vs. software • Differences between types of computer languages • Evolution of C++ • Computer and Information Ethics • Compiling a simple program • Compilation process • Algorithms	1 week	*For Support:* • choice to work with others or alone • visual demonstrations, illustrations, and models • work with checklists *For Enhancement:* • higher expectations in assessment questioning • extended research and readings	*Standards:* CRP1, CRP8, CRP11, 8.1.12.F.1, 8.2.12.E.4	*Formative Assessment:* • Worksheets (2): (PowerPoint- • Based) • Topic (5 facts on History of the Computer (as assigned) *Summative Assessment:* • PowerPoint Project - History, languages, careers, internet, security, ethics • Quiz (History of Computers)

Unit Topic	Time Allocated	Differentiating Instruction for Students with Disabilities, Students at Risk of School Failure, English Language Learners, & Gifted & Talented Students	Standards	Assessments
Numbers and Objects • Input/Output/Header files • Write Code in C++ • Data Types and Output/Format • Number Types • Input • Assignment Statements • Constants • Arithmetic • Functions as Black Boxes • Implementing Functions • Function Comments • Return Values • Parameters • Reference Parameters • Higher Level Mathematical Functions	2 weeks	*For Support:* • study guides • PowerPoint reference • partnering • guided notetaking • reteach and review *For Enhancement:* • flexible grouping on challenging exercises • higher level questioning techniques • facilitating classmates • independent extensions based on student interest, curiosity, and choice	*Standards:* CRP1, CRP6, 8.1.12.F.1, 8.2.12.E.4	*Formative Assessment:* • Questions of the Day using Google Classroom - based on current or previously learned content • Worksheets to reinforce understanding • Correcting code to problem solve *Summative Assessment:* • Lab#1 (User input and calculations) • Lab #2 (Higher level mathematical calculations with functions) • Test (Sec 2.1-2.5) (Numbers and Objects) • Mini-Quizzes (Corrections in Code)

Unit Topic	Time Allocated	Differentiating Instruction for Students with Disabilities, Students at Risk of School Failure, English Language Learners, & Gifted & Talented Students	Standards	Assessments
Strings • Variables • Substrings • Concatenation • Formatted Output	2 weeks	*For Support:* • PowerPoint reference • peer mentoring on lab • visual demonstrations *For Enhancement:* • more complex tasks and problems • higher expectation for writing programs	*Standards:* CRP6, CRP8 8.1.12.B.2, 8.1.12.D.5, 8.1.12.F.1	*Formative Assessment:* • Questions of the Day using Google Classroom - based on current or previously learned content • Applications of String methods *Summative Assessment:* • Sec (2.6) (Strings) Notes • Sample method practice from CodingBat (Strings) • Lab#3 (String Programs) • Test (Strings)
Relational Operators • The If Statement • Relational Operators • Multiple Alternatives • Nested Branches • Boolean Operations	4 weeks	*For Support:* • PowerPoint reference • demonstrations on smartboard • choice to work with others or alone • multilevel activities *For Enhancement:* • more complex tasks and problems • display and explain version of coded task • peer mentoring	*Standards:* CRP6, CRP8 8.2.12.E.3, 8.2.12.E.4	*Formative Assessment:* • Questions of the Day using Google Classroom - based on current or previously learned content • Truth Tables (3 sets) • Sample method practice from CodingBat (Logic) *Summative Assessment:* • Lab#4 (Logic) • Test (3.1-3.5) (If/Else/NestedIfs)

Unit Topic	Time Allocated	Differentiating Instruction for Students with Disabilities, Students at Risk of School Failure, English Language Learners, & Gifted & Talented Students	Standards	Assessments
Loops • The While Loop • The For Loop • The Do Loop • Nested Loops	7 weeks	*For Support:* • PowerPoint reference • guided notetaking • peer mentoring on lab • demonstrations on smartboard *For Enhancement:* • more complex tasks and problems • flexible grouping on challenging exercises • display and explain version of coded task • peer mentoring • curriculum compacting	*Standards:* CRP1, CRP8, CRP11, 8.1.12.F.1, 8.2.12.E.4	*Formative Assessment:* • Questions of the Day using Google Classroom - based on current or previously learned content • Worksheets (while / doWhile / for loops - 5) • Worksheets (nested loops - 2) *Summative Assessment:* • Sample programs from CodingBat • Lab#5 (while / doWhile loops • While / doWhile Quiz • Lab#6 (for loops) • Test (3.6-3.8) (while / doWhile / for loops)

Unit Topic	Time Allocated	Differentiating Instruction for Students with Disabilities, Students at Risk of School Failure, English Language Learners, & Gifted & Talented Students	Standards	Assessments
Functions • Side Effects • Random Numbers • Simulated Gaming Programs	2 weeks	*For Support:* • PowerPoint reference • visual demonstrations, illustrations, and models • activity choice *For Enhancement:* • more complex tasks and problems • higher expectations in assessment questioning	*Standards:* CRP6, CRP8 8.1.12.B.2, 8.2.12.E.3, 8.2.12.E.4	*Formative Assessment:* • Questions of the Day using Google Classroom - based on current or previously learned content • Worksheets Random numbers (2) *Summative Assessment:* • Lab#7 (Random numbers / Nested Loops) • Benchmark Q2
Arrays • Removing and Inserting Array • Elements • Reverse Orders • Searching / Sorting Orders	2 weeks	*For Support:* • work with checklists • guided notetaking • peer mentoring on lab *For Enhancement:* • flexible grouping on challenging exercises • higher level questioning techniques • facilitating classmates • higher expectation for writing programs	*Standards:* CRP2, CRP4 8.1.12.B.2, 8.1.12.D.5	*Formative Assessment:* • Questions of the Day using Google Classroom - based on current or previously learned content • Worksheets (arrays - 2) • Practice exercises with arrays from codingBat *Summative Assessment:* • Lab#8 (Arrays)