BUR OAK SECONDARY SCHOOL

Introduction to Computer Science

COURSE CODE: ICS3U

MINISTRY CURRICULUM DOCUMENT: ***The Ontario Curriculum, Grades 10 to 12: Computer Studies, 2008 (revised)***

MINISTRY PREREQUISITE: None **CREDIT VALUE: 1.0**

DEPARTMENT: Computer Studies **DEPARTMENT HEAD: Mr. Fernandes**

COURSE DESCRIPTION

This course introduces students to computer science. Students will design software independently and as part of a team, using industry-standard programming tools and applying the software development life-cycle model. They will also write and use subprograms within computer programs. Students will develop creative solutions for various types of problems as their understanding of the computing environment grows. They will also explore environmental and ergonomic issues, emerging research in computer science, and global career trends in computer-related fields.

MINISTRY LEARNING EXPECTATIONS

A. Programming Concepts and Skills

A1. Demonstrate the ability to use different data types, including one-dimensional arrays, in computer programs

A2. Demonstrate the ability to use control structures and simple algorithms in computer programs

A3. Demonstrate the ability to use subprograms within computer programs

A4. Use proper code maintenance techniques and conventions when creating computer programs

B. Software Development

B1. Use a variety of problem-solving strategies to solve different types of problems independently and as part of a team

B2. Design software solutions to meet a variety of challenges

B3. Design algorithms according to specifications

B4. Apply a software development life-cycle model to a software development project

C. Computer Environments and Systems

C1. Relate the specifications of computer components to user requirements

C2. Use appropriate file maintenance practices to organize and safeguard data

C3. Demonstrate an understanding of the software development process

D. Topics in Computer Science

D1. Describe policies on computer use that promote environmental stewardship and sustainability

D2. Demonstrate an understanding of emerging areas of computer science research

D3. Describe postsecondary education and career prospects related to computer studies

Course Units

Unit 1: Computer Science Fundamentals

Part A: Students are first introduced to the basic idea of computer science through media presentations and discussions. Next, the students are presented with topics on understanding basic hardware and software concepts and how they relate to programming. Then the students are given an overview of programming languages to become familiar with different types of paradigms. In particular, students are introduced to a typical Java development environment.

Part B: Students are taught to write simple, teacher-led Java programs in the form of simple Console and GUI based applications. Next, students review basic problem-solving methods and techniques and are introduced to types of errors they may encounter while coding. Then students begin to code their own basic programs in a series of activity-based lessons. These will include simple input, processing and output (IPO) programs with the use of basic data types allowing them to progress through the beginning phases of the programming language. Throughout the unit the students are presented with a particular focus on the terminology and paradigm of the language used throughout the course. Further, students will learn how to use an operating system in conjunction with a network to perform tasks such as file management.

Unit 2: Control Structures

This unit begins by reviewing basic problem-solving techniques and students are introduced to pseudocode and various visual design tools (flowcharts, activity diagrams, UML) in order to develop algorithms. To aid in this process the concept of top-down, stepwise refinement is explained. Students then explore the various coding structures used to control the flow of logic in a computer program. Concepts presented include the use of selection statements to choose among alternative actions and repetition statements to execute sections of a program repeatedly. Students will be able to see when to use counter-controlled repetition and sentinel-controlled repetition. Lastly, the use of logical operators is presented to form more complex conditional expressions in control statements.

Unit 3: Modular Programming and Data Structures

Part A: Students are presented with the basic concepts of methods being an essential part of creating modular programs to solve complex tasks - 'divide and conquer'. Students will see how the method call/return mechanism works and how it relates to a computer's internal memory. They will also gain a sense of how packages group related classes and explore the Java API. Next, using random-number generation they will implement game-playing applications. Lastly, they will explore the concept of scope and how it affects computer programs.

Part B: Students are first reminded of how arrays are simple structures that are common in our world - they are lists. Next, using this context students are introduced to the idea of using arrays as a basis for manipulating larger quantities of data. Then students are given a series of activity-based lessons that allow them to gain confidence in utilizing arrays in a variety of scenarios.

Part C: Students are being presented with computer programs being a tool for modelling 'real-world' objects. Students are taught the basic relationship between objects and classes. Students learn to define their own objects in terms of its attributes and behaviours. Then students will create programs using constructor methods to initialize objects. Students will also gain an appreciation for the differences between primitive and reference types. Lastly, students will create programs to utilize the objects they have created.

Unit 4: Software Development

Students are presented with the theory behind the Software Development Life Cycle. They will learn about the roles, tasks and deliverables of the phases and show this by creating a software development project. Also, this unit requires students to utilize a variety of other skills such as problem solving, project management and their ability as a member of a team.

Unit 5: Final Summative Project

Students demonstrate overall mastery of the major programming concepts presented in this course. They will select a larger programming task to apply their skills. Also, this unit requires students to utilize a variety of other skills such as problem solving, project management and their ability to attain needed skills independently.

Unit 6: Impact of Computers

This unit focuses on various topics relating to the impact of computers on society. Topics are introduced both throughout the course and as a dedicated section near the conclusion of the course. Topics of study include: the impact of computers on the environment and human health; social and economic impacts of computers on society; emerging computer technologies; and, current trends in computer careers along with essential skills/work habits and post-secondary pathway opportunities. In addition to teacher-led lessons, students will conduct independent research to find out current information that relates these ideas. Also, students will have opportunities to share their findings in a variety of ways that allows them to express their creativity, communication and literacy skills.

Teaching/Learning Strategies

A variety of teaching and learning strategies are used, including:

- Brainstorming – group generation of initial ideas to encourage participation and inclusion
- Collaborative/Cooperative Learning – small group learning providing high levels of student engagement, interdependence and inclusion
- Conferencing – student to student conversation and student to teacher conversation
- Kinesthetic activities - to engage other types of body learning
- Pair Programming - used to both learn and create with partners encouraging communication and sharing
- Problem-Based approach – using a series of steps to solve a problem
- Project-Based approach - students work on comprehensive projects that involve design, planning and testing
- Scaffolding - to build upon concepts in a strategic manner
- Role playing - used to better understand concepts by placing the student in various viewpoints
- Independent Study – students explore and research a topic of interest
- Inquiry – active learning and discovery through posing questions and researching answers
- Report/Presentation – oral and written presentation of researched topic to class
- Teacher modelling/student practice - provide opportunities to students to develop individual confidence
- Whole Group Instruction - used for shared understanding often as an introduction or summary of concepts

Assessment And Evaluation

The primary purpose of assessment and evaluation is to improve student learning. **The Achievement Chart for Computer Studies** will guide all assessment and evaluation.

The **final grade** will be determined as follows:

- **70% based on Assessment of Learning** conducted throughout the course
- **30% based on a Final Summative Project and/or a Final Exam** administered towards the end of the course

Category	Percent
Knowledge and Understanding	**25%**
Thinking and Inquiry	**25%**
Communication	**25%**
Application	**25%**
Total	**100%**

Assessment and evaluation is divided into two important parts: a) the grade the student receives on a midterm or final report indicates achievement/ proficiency in Curriculum Expectations, and b) a level of competency that will be assessed and reported in the following areas of Learning Skills and Work Habits: Independent Work, Collaboration, Responsibility, Initiative, Self-Regulation, and Organization. **See the Bur Oak Secondary School Assessment, Evaluation and Communication Policy on Google Classroom

Assessment for/as Learning

- Checklists – for formative teacher/peer/self-assessment
- Computer Programs – focusing on both process, communication, understanding and a final product
- Concept Maps - to make connections of various ideas and create a framework of understanding
- Conversations - discussion between students and/or teacher to share understanding and thought processes
- Demonstrations – show how students are learning and working
- Descriptive Feedback - anecdotal comments with suggestions for improvement
- Observations – seeing how students deal with problems and work in groups to solve problems
- Oral Question and Answer sessions - feedback for both students and teacher regarding understanding
- Performance Tasks - on-computer skill demonstrations
- Quizzes – feedback for both the student and the teacher about a few chosen expectations
- Reflection – tool to encourage students to be more involved in their own learning process
- Rubrics – provide clear expectations of performance at the start of an activity
- Self/Peer Assessment - students take ownership of their learning

Assessment of Learning

- Presentations - involving both oral and/or visual components
- Performance Tasks - on-computer skill demonstrations
- Research Projects - presented in various forms
- Software Projects – focussing on both process and the final product
- Triangulation - using conversations, observations and products together
- Unit Tests – paper and pencil tests used for assessing a variety of skills

Assessment & Evaluation of Student Learning Skills

- Teacher evaluations based on observations
- Student self-assessment
- Student-teacher conferences
- Self-reflection exercises

ACCOMMODATIONS

Assessment, instructional and environmental **accommodations** are provided to individual students as per their **IEP**. Similarly, **adaptations** for **Multi-Language Learners** are provided based upon the student's level of language development, strengths and needs.

The following are general accommodation/adaptation strategies used in this course:

- References and inclusion of recommendations from student IEPs and other records
- Provide adaptive hardware devices and/or software tools (e.g. large screen monitors, personal laptops, specialized software for various needs including electronic texts for easier translation and adoption)
- Provide appropriate environmental accommodations for students with various challenges
- Conferencing with Special Education Staff and students to discuss accommodations to ensure that various aspects of the classroom environment meet the needs of the students to fully participate in the program
- Conferencing with MLL Staff and students to discuss accommodations including providing translations of course content, lists of terms, and other resources as needed
- Strategic grouping of students to encourage peer interaction, support and development
- Flexible scheduling to organize and complete assigned tasks
- Provide examples and templates to give students a more clear framework as needed
- Assist students in dividing larger tasks into smaller more manageable tasks (chunking)
- Provide oral explanations and individual/small group conferencing to ensure understanding of concepts
- Provide alternative selection of problems (for example, adjusting context to a more familiar topic)
- Use of visual aids/manipulatives to assist students as needed
- Provide various enrichment opportunities to enhance learning

CONSIDERATION FOR BOARD INITIATIVES

Throughout the course, consideration will be given to incorporating the following board-wide initiatives into the content, processes and environment:

- Dismantling Anti-Black Racism Strategy
- Indigenous Education and Equity
- Mathematics
- Mental Health
- Modern Learning

RESOURCE MATERIALS

Students will have access to the following resources:

- Computer lab
- Java Programming Language
- Eclipse IDE
- Electronic text, presentations and notes
- Google Classroom and Apps
- Other hardware/software tools as needed
  **See the YRDSB Information Technology Acceptable Use Agreement on Google Classroom