

	Ontario Ministry of Education www.edu.gov.on.ca/eng/		Toronto District School Board www.tdsb.on.ca		R.H. KING ACADEMY http://schools.tdsb.on.ca/rhking/
COURSE OF STUDY OUTLINE					
Department	<i>Information Technology</i>	Course Type	<i>Open</i>		
Teacher	<i>Pavel Muresan</i>	Grade	<i>10</i>		
Course Title	<i>Introduction to Computer Studies</i>	Credit Value	<i>One</i>		
Course Code	<i>ICS20</i>	Prerequisites	<i>None</i>		
Ministry Document	<i>The Ontario Curriculum. Grades 10 to 12. Computer Studies. Ontario, 2008. http://www.edu.gov.on.ca/eng/curriculum/secondary/computer10to12_2008.pdf</i>				
Learning Resources	<i>Course Electronic folder, Textbook, Student binder, Course Web site, Visual Studio. VB.Net Programming Environment</i>				

ICS20 COURSE: INTRODUCTION TO COMPUTER STUDIES

COURSE OVERVIEW – INTRODUCTION TO COMPUTER STUDIES, GRADE 10, OPEN (ICS20)

- 1. MINISTRY OF EDUCATION CURRICULUM POLICY DOCUMENT:** The Ontario Curriculum. Grades 10 to 12. Computer Studies. Ontario, pp.33-38. ISBN 978-1-4249-8102-1. © Queen's Printer for Ontario, 2008.
- 2. TDSB document:** Computer Studies. Program Guide. Revised Curriculum 2009. Resources developed by teachers for teachers. TDSB School services: Program – ICT, e-learning and Computer Studies. Web site: tdsbweb/program/computerstudies
- 3. Web Sites:** www.edu.gov.on.ca/eng/curriculum/secondary/subjects.html

COURSE DEVELOPED: September 2016

RESOURCES:

- 1. Textbook:** Beth Brown, Bruce Presley. *An Introduction to Programming Using Microsoft Visual Basic.Net*, Lawrenceville Press, 2003. ISBN 1-58003-038-6.
- 2. Computer Lab A22: Visual Studio. VB.Net Programming Environment**
- 3. Student Binder** (15 dividers plus cover page) containing all handouts and assignments.
- 4. Course Folder in the TDSB File Server.**

A. COURSE DESCRIPTION/RATIONALE

This course introduces students to computer science concepts and programming. Students will plan and write simple computer programs by applying fundamental programming concepts, and learn to create clear and maintainable internal documentation. They will also learn to manage a computer by studying hardware configurations, software selection, operating system functions, networking, and safe computing practices. Students will also investigate the social impact of computer technologies, and develop an understanding of environmental and ethical issues related to the use of computers. Students also develop an awareness of computer-related careers and the impact of computers and associated technologies. (Computer Studies: The Ontario Curriculum Grades 10 to 12, Revised, 2008). This course is designated as open and can be taken by all students who wish to learn about Computer and Information Science. Students who wish to continue study in this area can take the is the Major Project, where students are required to design, write code, test and debug and run a large fully functional Visual Basic application. For this course each student has to have a binder with dividers for each module. The binder has to contain all the lessons, notes, handouts, assignments, tests and quizzes, and it will be marked twice: midterm and at the end of the course.

UNIT SUMMARY:

Unit Title	Hours
<p>Unit 1: Hardware and Software This unit will focus on the functions of internal and external hardware components as well as the types of and differences between application software and operating systems. It will also address the hardware and software needs of various users.</p>	15
<p>Unit 2: Operating Systems and Maintenance This unit will address the similarities/differences and benefits of various operating systems (Windows, XO, Linux, etc.) and the importance of system maintenance in view of virus protection and software updates.</p>	6
<p>Unit 3: Networking Computers This unit will allow students to identify the various requirements necessary to operate a home computer network. Students will identify the software for communications, hardware for physical setup, and methods for sharing resources in a home environment.</p>	11
<p>Unit 4: Career and Post-Secondary Opportunities Students will research and describe the various career opportunities available to students interested in computer studies and the educational pathway and essential skills necessary to prepare for post-secondary education and the world of work.</p>	15
<p>Unit 5: Programming Concepts This unit will focus on input, output, data types, basic and built in math functions, syntax and logic errors and run time errors.</p>	10
<p>Unit 6: Programming Structures. This unit will focus on Boolean operators, selection statements and repetition constructs. Inclusion of graphics in this unit may help to motivate and engage students, but the concepts will not be evaluated.</p>	12
<p>Unit 7: Developing an Application Program. This unit will focus on combining all previous programming units into student-led activities.</p>	12
<p>Unit 8: Computers and Society Students will explore and explain the impact of computers and related technology on society and the environment, acceptable use policies, and the legal and ethical issues related to technology use.</p>	15
<p>Unit 9: Culminating Activity – Project It's summer vacation, you have all your volunteer hours so you have decided to start your own business and make some money in the convenience of your own home. To cut down on overhead costs you have decided to become a member of the e-world of business called s-Bay, a popular electronic school-based vendor. Accordingly, you will design and create a program that will allow patrons to purchase items from your store (the program should be similar to the <i>Fast Lane</i> found at movie theatres).</p>	14
<p>Culminating Exam This should evaluate all overall expectations not addressed in the above culminating activity.</p>	

UNIT TITLE , DESCRIPTION, AND EXPECTATION (SEQUENCE AND TIME)

UNIT/MODULE	UNIT OVERALL AND SPECIFIC EXPECTATIONS	HOURS
<p>COMPUTERS, ENVIRONMENT AND SOCIETY. ERGONOMICS AND ETHICAL ISSUES</p>	<p>Students will explore and explain the impact of computers and related technology on society and the environment, acceptable use policies, and the legal and ethical issues related to technology use. Overall Expectations: C1. describe key aspects of the impact of computers and related technologies on society; C2. describe computer use policies that promote environmental stewardship and sustainability; C3. describe legal and ethical issues related to the use of computing devices. Specific Expectations: C1.1 describe a variety of adaptive technologies that help to improve computer accessibility (e.g., text-to-speech, speech-to-text, adapted mouse, font control, ergonomic keyboard, virtual</p>	15

	<p>keyboard, sticky keys, colour contrast, image magnifier); C1.2 explain the impact on privacy of techniques for collecting and processing data (e.g., camera phones, reward programs, targeted advertising, digital rights management, monitoring software); C1.3 describe how portable computing devices (e.g., PDA, cell phone, GPS, laptop) affect our everyday lives; C1.4 describe how electronic access to information (e.g., instant messaging, webcasts, social networking sites, wikis, blogs, video sharing sites) influences our everyday lives, as well as the lives of people in various countries around the world, in both positive and negative ways; C1.5 describe issues associated with access to online services (e.g., reliability of passwords, network security, identity theft, the permanence of information released onto the Internet); C2.1 describe the negative effects of computers and computer use on the environment (e.g., chemicals from electronic waste dumped in landfills – domestic or overseas – leaching into soil and groundwater; unnecessary use of paper; heavy power consumption) and on human health (e.g., effects of exposure to radiation, musculoskeletal disorders, eye strain, mental health and behavioural problems created or exacerbated by social isolation); C2.2 identify measures that help reduce the negative effects of computers on the environment (e.g., lab regulations, school policies, corporate policies, provincial policies, paperless workplaces) and on human health (e.g., ergonomic standards); C2.3 describe ways in which computers are or could be used to reduce resource use and to support environmental protection measures (e.g., computer modelling to reduce use of physical resources; interpretation of large amounts of environmental data; management of natural resources; programmable temperature control to reduce energy consumption); C2.4 describe, on the basis of research, how and where recycled electronic waste is processed, and identify local companies and institutions that offer such services; C3.1 describe legal and ethical issues related to the use of computers (e.g., music and video file downloading, spyware, identity theft, phishing, keystroke logging, packet sniffing, cyber bullying); C3.2 describe safeguards (e.g., effective passwords, secure websites, firewalls, biometric data) for preventing the unethical use of computers.</p>	
<p>PC HARDWARE AND SOFTWARE</p>	<p>This unit will focus on the functions of internal and external hardware components as well as the types of and differences between application software and operating systems. It will also address the hardware and software needs of various users. Overall Expectations: A1. describe the functions of different types of hardware components, and assess the hardware needs of users; A2. describe the different types of software products, and assess the software needs of users. Specific Expectations: A1.1 use correct terminology to describe computer hardware (e.g., USB, FSB, IEEE 1394 interface), speed measurements (e.g., megahertz), and size measurements (e.g., megabytes, gigabytes); A1.2 describe the functions of the internal components of a computer (e.g., CPU, RAM, ROM, cache, hard drive, motherboard, power supply, video card, sound card); A1.3 describe the functions of common computer peripheral devices (e.g., printer, monitor, scanner, keyboard, mouse, speakers, USB flash drive); A1.4 assess user computing needs and select appropriate hardware components for different situations (e.g., a student on a fixed budget, a home business user, a gaming enthusiast, a photographer, a home video enthusiast, a distance education user, a human resources manager, an accountant); A2.1 explain the difference between software used for applications (e.g., word processor, spreadsheet, email client), programming (e.g., an integrated development environment), and systems (e.g., operating system tools such as a registry editor and a defragmenting tool); A2.2 assess user computing needs and select appropriate software for different situations (e.g., a student on a fixed budget, a home business user, a gaming enthusiast, a photographer, a home video enthusiast, a distance education user, a human resources manager, an accountant).</p>	<p>15</p>
<p>OPERATING SYSTEMS, MAINTENANCE AND SECURITY</p>	<p>This unit will address the similarities/differences and benefits of various operating systems and the importance of system maintenance in view of virus protection and software updates. Overall Expectations: A3. use the basic functions of an operating system correctly; A5. explain the importance of software updates and system maintenance to manage the performance and increase the security of a computer. Specific Expectations: A3.1 describe operating system functions that meet various user needs (e.g., running applications, organizing files, managing users, configuring peripherals); A3.2 use file management techniques to organize and manage files (e.g.,</p>	<p>6</p>

	copy, move, delete, rename files; create shortcut); A3.3 use general keyboard shortcuts to perform common tasks (e.g., cut, copy, paste, print, print window, print screen); A3.4 describe the features and limitations of various operating systems; A5.1 describe different types of malware (e.g., viruses, Trojan horses, worms, spyware, adware, malevolent macros) and common signs of an intrusion, and explain how to prevent malware attacks; A5.2 explain the importance of maintaining software updates (e.g., operating system updates, application software updates, virus definitions) to increase computer security and maintain hardware and software compatibility; A5.3 explain the importance of preventive maintenance (e.g., defragmenting a hard drive, deleting unused software and data files) to manage computer performance.	
NETWORKING SYSTEMS	This unit will allow students to identify the various requirements necessary to operate a home computer network. Students will identify the software for communications, hardware for physical setup, and methods for sharing resources in a home environment. Overall Expectations: A4. demonstrate an understanding of home computer networking concepts. Specific Expectations: A4.1 identify various networking applications and protocols (e.g., VoIP, streaming media, FTP, email, instant messaging); A4.2 describe the features and functions of wired and wireless networking hardware (e.g., NICs, routers, hubs, cables, modems); A4.3 demonstrate an understanding of various methods for sharing network resources (e.g., shared file access, shared printer access, Internet access).	11
COMPUTER CAREERS AND POSTSECONDARY OPPORTUNITIES (ISU)	Students will research and describe the various career opportunities available to students interested in computer studies and the educational pathway and essential skills necessary to prepare for post-secondary education and the world of work. Overall Expectations: C4. describe postsecondary education and career prospects related to computer studies. Specific Expectations: C4.1 research and describe trends in careers that require computer skills, using local and national sources (e.g., local newspaper, national newspaper, career websites); C4.2 research and report on postsecondary educational programs leading to careers in the field of information systems and computer science (e.g., institutions offering relevant programs, industry certifications, courses of study, entrance requirements, length of programs, costs); C4.3 identify groups and programs that are available to support students who are interested in pursuing non-traditional career choices in computer-related fields (e.g., mentoring programs, virtual networking/support groups, specialized postsecondary programs, relevant trade/industry associations); C4.4 identify the Essential Skills and work habits that are important for success in computer studies, as defined in the Ontario Skills Passport.	15
PROBLEM SOLVING AND FLOW CHARTS	This unit focuses on how to program a computer using a problem-solving model . This model helps to organize and develop the fundamental structures of programming. Students learn to apply a specific problem-solving model to a diverse set of problems and to develop general problem-solving skills through application of the software design process in a project setting. To emphasize that problem solving is both a team skill and an individual effort, groups are involved in discussions, brainstorming sessions, and co-operative learning situations. The end-of-unit assessment is based on a group project and focuses on the process of the group as well as the product. Students build awareness of computer-based careers by exploring career information available through the school and local community, and by sharing of information using available computer media such as web pages and presentation software. 1. Software Development Life Cycle (SDLC). 2. Problem Solving Models. 3. Flow Charts. 4. Algorithms. 5. Pseudo-code.	6
PROGRAMMING CONCEPTS. VISUAL BASIC.NET IDE	This unit introduces Visual Basic Programming language and Integrated Development Environment (IDE) 1. Visual Basic Programming Language. 2. A Visual Basic Application. 3. The Visual Basic IDE. 4. Adding Objects to a Form. 5. Object Property Values. 6. Resizing and Moving an Object. 7. Saving a Project. 8. Running a Visual Basic Application. 9. Objects and their Event Procedures. 10. Using Assignment to Change Property Values. 11. The Form_Load Event Procedure. 12. Commenting Code. 13. Operators and Expressions. 14. Creating an Executable File.	4
PROGRAMMING CONCEPTS: VARIABLES	This unit focuses on basic programming concepts: data types, basic and built in math functions, variable declarations, assignment statements, input/output, syntax and logic errors Overall Expectations: B1. describe fundamental programming concepts and	8

AND CONSTANTS	<p>constructs; B2. plan and write simple programs using fundamental programming concepts; B3. apply basic code maintenance techniques when writing programs.</p> <p>Specific Expectations: B1.2 describe the types of data that computers can process and store (e.g., numbers, text); B1.3 explain the difference between constants and variables used in programming; B1.4 determine the expressions and instructions to use in a programming statement, taking into account the order of operations (e.g., precedence of arithmetic operators, assignment operators, and relational operators); B2.2 use variables, expressions, and assignment statements to store and manipulate numbers and text in a program; B2.3 write keyboard input and screen output statements that conform to program specifications; B2.6 explain the difference between syntax, logic, and run-time errors; B3.1 write clear and maintainable code using proper programming standards (e.g., indentation; naming conventions for constants, variables, and expressions); B3.2 write clear and maintainable internal documentation to a specific set of standards (e.g., program header: author, revision date, program name, program description; table of variable names and descriptions); B3.3 use a tracing technique to understand program flow and to identify and correct logic and run-time errors in a computer program. Topics to be covered: 1. Using Variable. 2. Variable Assignment. 3. Using Name Constants. 4. Choosing Identifiers. 5. Built-In Data Types. 6. Variable Declarations. 7. Syntax Errors. 8. Debugging Techniques. 9. Obtaining a Value from the User. 10. Automatic Type Conversion. 11. Special Division Operators. 12. Option Buttons. 13. Visual Basic Programming Guidelines. 14. Case Study.</p>	
PROGRAMMING CONCEPTS: DECISION STRUCTURES	<p>This unit introduces if statements as decision/selection structure The unit will focus on Boolean operators and selection statements. Inclusion of graphics in this unit may help to motivate and engage students, but the concepts will not be evaluated. Overall Expectations: B1. describe fundamental programming concepts and constructs; B2. plan and write simple programs using fundamental programming concepts; B3. apply basic code maintenance techniques when writing programs. Specific Expectations: B1.5 identify situations in which decision and looping structures are required; B1.6 describe the function of Boolean operators (e.g., AND, OR, NOT), comparison operators (i.e., equal to, not equal to, greater than, less than, greater than or equal to, less than or equal to), and arithmetic operators (e.g., addition, subtraction, multiplication, division, exponentiation, parentheses), and use them correctly in programming; B2.1 use a visual problem-solving model (e.g., IPO [Input, Process, Output] chart; HIPO [Hierarchy plus Input, Process, Output] chart and diagram; flow chart; storyboard) to plan the content of a program; B2.4 write a program that includes a decision structure for two or more choices (e.g., guessing game, rock-paper-scissors game, multiple-choice quiz, trivia game). Topics to be covered: 1. The If... Then Statement. 2. Round off Error. 3. The If...Then...Else Statement. 4. Nested If...Then...Else Statements. 5. Generating Random Numbers. 6. Scope. 7. Logical Operators. 8. Algorithms. 9. Message Boxes. 10. Creating a Password Application. 11. Using Counters. 12. Check boxes. 13. Printing a form. 14. Visual Basic Programming Guidelines.</p>	8
PROGRAMMING CONCEPTS: LOOPS AND FUNCTIONS	<p>This unit focuses on repetition or iteration programming structure. Students develop their programming skills through practical programming activities that incorporate fundamental program constructs like repetitive structures. Overall Expectations: B1. describe fundamental programming concepts and constructs; B2. plan and write simple programs using fundamental programming concepts; B3. apply basic code maintenance techniques when writing programs. Specific Expectations: B2.5 write programs that use looping structures effectively (e.g., simple animation, simple board games, coin toss); B2.7 compare and contrast the use of different programming environments to solve the same problem (e.g., a solution developed in a programming language versus one developed using a spreadsheet); Topics to be covered: 1. Do...Loop Statement. The programming software introduced in this module allows students to write simple programs which integrates with hardware to control external devices and peripherals. 2. Infinite Loops. 3. Input boxes. 4. Using Accumulators. 5. String Conversion Functions. 6. Manipulating Strings. 7. The Len function. 8. The InStr function. 9. For...Next Statement. 10. Generating Strings. 11. Character data Storage. 12. Comparing Strings. CASE STUDY.</p>	8
MAJOR PROJECT	In this culminating activity, students demonstrate overall mastery of the course content	

	and expectations. Students perform hardware and networking demonstrations of skills, and apply knowledge and skills in problem-solving, programming, hardware, and networking. They demonstrate the ability to design, create, test, and debug computer programs. Design, write code and test a fully functional VB Application	
MAJOR PROJECT PRESENTATIONS	Seven to 10 minutes for each student.	

TEACHING/LEARNING STRATEGIES

All units are activity-based. Socratic lessons, teacher demonstrations, and research activities provide students with the necessary terminology and methodology necessary to complete the various activities. Classroom discussions, brainstorming, and collaborative and/or co-operative learning is used to assist students in meeting course expectations. Problem-solving exercises are used. Students also research, write reports, and take notes in meeting expectations. Upon completion of this course, students demonstrate the ability to apply skills and knowledge to practical work tasks that involve planning and implementation processes, completion of work assignments, and various problem-solving activities.

OVERVIEW OF EVALUATION AND WEIGHTING

Students are provided with opportunities to demonstrate the highest level of their achievement of the expectations in the four achievement categories. Students will be evaluated based on the following four categories of knowledge and skills in technological education (70% of the final mark) and a final exam (30% of final mark). Students will be graded based on the following categories:

Knowledge/Understanding	15%
Thinking and Inquiry	20%
Application	20%
Communication	15%
Final Exam (20%) Major Project. Culminating activity (10%)	10% + 20% = 30%

COURSE EVALUATION PLAN

70% Course Work

(K/U = Knowledge/Understanding; TI = Thinking and Inquiry; A = Application; C= Communication)

CHAPTER/UNIT	EVALUATION TASK	ACHIEVEMENT CHART FOCUS	WEEK ENDING	DUE DATE
COMPUTER, ENVIRONMENT AND SOCIETY. ERGONOMICS AND ETHICAL ISSUES	Assignment: Design your Ergonomic Office. Ergonomics Quiz. Assignment: Essay/Report. Social Impact of IT.	K/U, TI, C, A	WEEK 1-2	
PC HARDWARE AND SOFTWARE	Assignment: Design your Computer Dream. Major Assignment: Computer Components. Test PC Hardware and Software. Practical Assignment: Build your own computer.	K/U, TI, A, C	WEEK 4-6	
OPERATING SYSTEMS, MAINTENANCE AND SECURITY	Quiz: Operating Systems	K/U, TI, A	WEEK 7	
NETWORKING SYSTEMS	Class Assignment: Network Topologies. Test: Networking	K/U, TI, A	WEEK 8	

COMPUTER CAREERS AND POSTSECONDARY OPPORTUNITIES (ISU)	Major Assignment: Computer Careers.	K/U, C	WEEK 9	
PROBLEM SOLVING AND FLOW CHARTS	Test: Flow Charts, Algorithms and Pseudo code	K/U, TI, C, A	WEEK 10	
PROGRAMMING CONCEPTS: VISUAL BASIC.NET IDE	Chapter 3 Test- Theory. Chapter 3 Test- Programming. Chapter 3 Exercises and Reviews	K/U, TI, A	WEEK 11	
PROGRAMMING CONCEPTS: VARIABLES AND CONSTANTS	Chapter 4 Test - Theory. Chapter 4 Test- Programming. Chapter 4 Exercises and Reviews	K/U, TI, A	WEEK 12	
PROGRAMMING CONCEPTS: DECISION STRUCTURES	Chapter 5 Test – Theory. Chapter 5 Test- Programming. Chapter 5 Exercises and Reviews	K/U, TI, A	WEEK 13	
PROGRAMMING CONCEPTS: LOOPS AND FUNCTIONS	Chapter 6 Test – Theory . Chapter 6 Test-Programming. Chapter 6 Exercises and Reviews	K/U, TI, A	WEEK 14 - 15	
MAJOR PROJECT BINDER EVALUATION	Major Project - Programming; Team Work	K/U, TI, C, A Learning Skills	WEEK 16-17	
PROJECT PRESENTATIONS	Major Project Presentations	K/U, TI, C, A	WEEK 18	
EXAM REVIEW		K/U, TI, C, A	WEEK 19	
EXAM	Final Exam	K/U, TI, A, C	WEEK 20	

30% Final Evaluations

Task	Achievement Chart Focus	Weighting
Examination	All categories: K/U, TI, C, A	30%

WEEKLY AGENDA

ICS20 COURSE: INTRODUCTION TO COMPUTER STUDIES, GRADE 10, OPEN

WEEK	MODUL	UNITS/LESSONS	EXPECTATIONS: PERFORMANCE INDICATORS
<u>WEEK 1</u>	WEB DESIGN AND E-PORTFOLIO	1. Web Site Architecture and Design; 2. E-portfolio hierarchical/circular model.	<i>By the end of this module, students will be able to:</i> design a fully functional Web Site using different Web tools; apply hierarchical and circular model in web design and navigation.
<u>WEEK 2</u>	ERGONOMICS	1. Health and safety problems associated with computer use 2. Safe practices in the handling of computer hardware and electronic	<i>By the end of this module, students will be able to:</i> use appropriate strategies to avoid potential health and safety problems associated with computer use, such as posture problems, eye strain, and musculoskeletal injuries; use safe practices in the handling of computer hardware and electronic components;

		components	
<u>WEEK 3</u>	COMPUTER TECHNOLOGY: SOCIAL IMPACT AND CONSEQUENCES	<p>1. The History of Computer Electronics and its Social Impact</p> <p>2. The Social and Ethical Implications of Computers</p> <p>3. Protecting Computer Software and Data.</p> <p>4. The Ethical Responsibilities of the Programmer.</p>	<p><i>By the end of this module, students will be able to:</i> describe the evolution of computer electronics; identify the social impact of computers and associated technologies; identify important scientific advances in computer electronic components; describe the development of computer engineering technology and its impact; describe how computer engineering has evolved and how it has affected people's security, safety, and privacy; demonstrate understanding of the importance of ethical computer use; demonstrate compliance with acceptable-use policies;</p>
<u>WEEK 4</u> <u>WEEK 5</u> <u>WEEK 6</u>	PC HARDWARE AND SOFTWARE	<p>1. Computer Architecture. 2. PC Components. 3. Input Devices. 4. Output Devices. 5. Communication Devices. 6. Motherboard, Slots and PC Cards. 7. CPU. 8. Primary Storage. RAM and ROM. 9. Secondary Storage Devices. How Disk Storage Works. How Floppy Drive and Hard Drive Work. 10. Storage Devices and RAID. 11. Ports and Connectors. How Computer Ports Work. 12. POST (Power-On-Self-Test). 13. BIOS and CMOS. 14. Multimedia. How CD-ROM Works. 15. Bus Architecture. How a Bus Works.</p>	<p><i>By the end of this module, students will be able to:</i> explain the historical development of a computer system as well as the relationship between computer internal hardware and software and external devices; describe a problem-solving model such as the input, processing, output model; identify the basic internal and external components of a computer; describe the primary function of each basic component; identify computer internals and peripheral devices and describe their relationship; connect and use correctly a variety of computer components and peripherals; use precise terminology in relation to all hardware, interfaces, and networking systems; set up a desktop computer system and install software; use safe practices in the handling of computer hardware and electronic components; demonstrate compliance with acceptable-use policies.</p>
<u>WEEK 7</u>	OPERATING SYSTEMS	<p>1. History of Operating Systems. 2. Operating System Components. 3. Operating System Interface. 4. Operating System Structure. 5. How an Operating System Controls Hardware.</p>	<p><i>By the end of this module, students will be able to:</i> describe how the internal components of the computer enable the peripherals to function; demonstrate the use of an operating system, including a network; describe the primary function of each basic component; use Internetworking services correctly to access and navigate global information resources; use appropriate file management techniques.</p>
<u>WEEK 8</u>	NETWORKING SYSTEMS	<p>1. Network Components. 2. Network Topologies: Mesh, Bus Star, Ring. 3. Network Devices: Networking Card, Modem, Hub, Switch, Bridge, Router, Gateway. 4. OSI Reference Model. 5. LAN, MAN, WAN. 6. Networking Protocols. 7. Peer -to- Peer Networking and Client- Server Networking. 8. IP Addresses and their Format. 9. TCP/IP Protocol. I</p>	<p><i>By the end of this module, students will be able to:</i> describe network types, topology, architecture, and cabling standards; use precise terminology in relation to all hardware, interfaces, and networking systems; perform or observe a basic connectivity task involving two computers, a parallel or serial cable, and external devices (e.g., peripherals);</p>
<u>WEEK 9</u>	COMPUTER CAREERS (ISU)	<p>1. Computer and IT related jobs. 2. Job Description. Nature of the work. 3. Skills set. 4. Training path. Qualification requirements. 5. Main Industries of Employment. 6. Related occupations . 7. Transferable Skills . 8. Trends and Outlooks. 9. Annual Average Income. 10. Resources.</p>	<p><i>By the end of this module, students will be able to:</i> identify and describe careers related to computer engineering; analyse the influences of computers on the engineering profession; identify computer skills that are important to employers.</p>
<u>WEEK 10</u> <u>WEEK 11</u>	PROBLEM SOLVING AND FLOW CHARTS	<p>1. Software Development Life Cycle (SDLC). 2. Problem Solving Models. 3. Flow Charts. 4. Algorithms. 5. Pseudo-code.</p>	<p><i>By the end of this module, students will be able to:</i> describe a problem-solving model such as the input, processing, output model; explain software process design (SDLC, PDLC, IPO charts, HIPO charts); explain internal numbering and character representation systems; draw flow charts for solving abstract and real life problems using control structures (sequential, decision, iteration);</p>

			write algorithms and pseudo-code.
<u>WEEK 12</u>	FOUNDATION OF PROGRAMMING CONCEPTS: VISUAL BASIC IDE	1. Visual Basic Programming Language. 2. A Visual Basic Application. 3. The Visual Basic IDE. 4. Adding Objects to a Form. 5. Object Property Values. 6. Resizing and Moving an Object. 7. Saving a Project. 8. Running a Visual Basic Application. 9. Objects and their Event Procedures. 10. Using Assignment to Change Property Values. 11. The Form_Load Event Procedure. 12. Commenting Code. 13. Operators and Expressions. 14. Creating an Executable File.	<i>By the end of this module, students will be able to:</i> describe the fundamental programming constructs; apply fundamental programming constructs to develop programs that interact with external components; describe the relationship between the binary number system and computer logic; define a standard way of representing characters in a binary code; use Visual Basic IDE; understand objects and add objects to a form; create a VB application; understand event procedures; change property values at design time and run time; add comments to a program; understand operators and expressions; create an executable file; understand good programming style guidelines.
<u>WEEK 13</u>	PROGRAMMING CONCEPTS: VARIABLES AND CONSTANTS	1. Using Variable. 2. Variable Assignment. 3. Using Name Constants. 4. Choosing Identifiers. 5. Built-In Data Types. 6. Variable Declarations. 7. Syntax Errors. 8. Debugging Techniques. 9. Obtaining a Value from the User. 10. Automatic Type Conversion. 11. Special Division Operators. 12. Option Buttons. 13. Visual Basic Programming Guidelines. 14. Case Study.	<i>By the end of this module, students will be able to:</i> define constants, variables, expressions, and assignments statements, including the order in which the operations are performed; describe how computers store and work with different types of data, including numbers and characters; use input and output statements in a program; declare variables in program code; understand variable assignment statements; use named constants; use different data types; declare multiple variable in program code; understand syntax errors including run-time errors; use text box objects in application; use change event procedures; understand automatic type conversion; use special division operators.
<u>WEEK 14</u>	PROGRAMMING CONCEPTS: DECISION STRUCTURES	1. The If... Then Statement. 2. Roundoff Error. 3. The If...Then...Else Statement. 4. Nested If...Then...Else Statements. 5. Generating Random Numbers. 6. Scope. 7. Logical Operators. 8. Algorithms. 9. Message Boxes. 10. Creating a Password Application. 11. Using Counters. 12. Check boxes. 13. Printing a form. 14. Visual Basic Programming Guidelines.	<i>By the end of this module, students will be able to:</i> use a decision structure in a program; use If...Then, If ...Then... Else and If...Then...Else statements; understand Nested If...Then...Else statements; generating random numbers; return integer portion of a number without rounding; understand the scope of a variable or constant; make global declarations; use And, Or, Not in Boolean expressions; understand algorithms and pseudocode; understand logic errors; use message boxes in applications; use counters in applications; use check box objects in applications; understand good programming style guidelines.
<u>WEEK 15</u> <u>WEEK 16</u>	PROGRAMMING CONCEPTS: LOOPS AND FUNCTIONS	1. Do...Loop Statement. 2. Infinite Loops. 3. Input boxes. 4. Using Accumulators. 5. String Conversion Functions. 6. Manipulating Strings. 7. The Len function. 8. the InStr function. 9. The for...Next Statement. 10. Generating Strings. 11. Character data Storage. 12. Comparing Strings. CASE STUDY.	<i>By the end of this module, students will be able to:</i> use a repetition structure in a program; design, write, and test a computer program to control a simple peripherals; understand Do...Loop statement and infinite loops; use input boxes in applications; understand accumulators and sentinels; understand string functions; understand For... Next statements and how to use Step; understand how the characters are stored in memory; understand the ASCII code and use the Asc and Chr functions in programs.
<u>WEEK 17</u>	WORKING ON MAJOR PROJECT	To apply programming and interfaces skills and knowledge.	<i>By the end of this module, students will be able to apply all basic programming and interfaces concepts and skills learned.</i>
<u>WEEK 18</u>	MAJOR PROJECT PRESENTATIONS	To prove communication skills and computer engineering competency and skills.	<i>By the end of this module, students will be able to present successfully the major project in front of the class proving communication skills and computer engineering competencies.</i>
<u>WEEK 19</u>	EXAM REVIEW		<i>By the end of this module, students will be able to write successfully TEE20 Exam applying all concepts, skills and values learned in this course.</i>
<u>WEEK 20</u>	EXAM	Available online also.	Global and synthetic indicator - 30% of final mark.

Learning Skills Evaluation Criteria

Learning Skills and Work Habits	Sample Behaviours. The student:
Responsibility	<ul style="list-style-type: none"> fulfils responsibilities and commitments within the learning environment; completes and submits class work, homework, and assignments according to agreed-upon timelines; takes responsibility for and manages own behaviour.
Organization	<ul style="list-style-type: none"> devises and follows a plan and process for completing work and tasks; establishes priorities and manages time to complete tasks and achieve goals; identifies, gathers, evaluates, and uses information, technology, and resources to complete tasks.
Independent Work	<ul style="list-style-type: none"> independently monitors, assesses, and revises plans to complete tasks and meet goals; uses class time appropriately to complete tasks; follows instructions with minimal supervision.
Collaboration	<ul style="list-style-type: none"> accepts various roles and an equitable share of work in a group; responds positively to the ideas, opinions, values, and traditions of others; builds healthy peer-to-peer relationships through personal and media-assisted interactions; works with others to resolve conflicts and build consensus to achieve group goals; shares information, resources, and expertise and promotes critical thinking to solve problems and make decisions.
Initiative	<ul style="list-style-type: none"> looks for and acts on new ideas and opportunities for learning; demonstrates the capacity for innovation and a willingness to take risks; demonstrates curiosity and interest in learning; approaches new tasks with a positive attitude; recognizes and advocates appropriately for the rights of self and others.
Self-regulation	<ul style="list-style-type: none"> sets own individual goals and monitors progress towards achieving them; seeks clarification or assistance when needed; assesses and reflects critically on own strengths, needs, and interests; identifies learning opportunities, choices, and strategies to meet personal needs and achieve goals; perseveres and makes an effort when responding to challenges.

Diagnostic evaluation: at the beginning of a term, a unit of study, or whenever information about prior learning is useful. Methods: unit pre-tests, skill inventory.

Formative evaluation: during learning, ongoing feedback to students of their strengths, weaknesses, and achievement of the expectations. Methods: course binder, course folder structure and updates, e-portfolio, self-assessment rubrics, checklists for programming problems, student/teacher conferencing, observation, peer assessment rubrics, check lists, quizzes, anecdotal comments with suggestions for improvement, self-assessment, journal, learning logs, reflection – important tools to encourage students to be more involved in their own learning process.

Summative evaluation: at the end of a learning process. Methods: classroom presentations, paper and pencil quizzes, tests, unit tests, final exam; assignments and projects evaluated using rubrics; culminating challenges; online tests, quizzes and exam; rubrics; skills demonstration – hardware and networking skill demonstration; computer programs – focusing on both process and the final product; research projects.

Assessment/Evaluation Techniques

The assessment includes the following: Paper and Pencil Tests – an appropriate means of assessing Theory and Foundation expectations; Unit tests – can be a combination of paper and pencil and on-computer skill demonstrations; Quizzes – feedback for both the student and the teacher about a few chosen expectations; Computer programs – focusing on both process and the final product; Research projects. General assessment tools include: Checklists – for formative teacher/peer/self assessment; Rubrics – provide clear expectations of performance at the start of an activity.

Tests/Quizzes

Tests will fall under the knowledge/understanding and thinking/inquiry achievement categories. Tests can be either theoretical or practical (performed on the computer) or a combination of both. Students will be informed well in advance of any test. Tests/Quizzes will be given on a regular basis. Students will be notified about tests/quizzes at least one day in advance. At the end of each unit a test will be given. Tests may be either

written or practical. Students will be notified about unit tests a week in advance. Performance on tests and quizzes will be used to assess the achievement of knowledge and skills. If a test is missed due to illness a doctor's note must be presented. **Students who are going to miss a test because of a school trip or team activity must notify their teacher two days prior to the test date, so that an earlier test date may be assigned for those students.** These are the only valid reasons for a rewrite. **Failure to write a test will result in a NOT COMPLETE.** A student may only make up one test during the semester. Any other missed test will result in a **NOT COMPLETE.**

Missed Tests/Quizzes: if a student knows they will be absent on the day of a test, they must notify the teacher in advance. Arrangements will be made for the student to write the test at an alternate time if an acceptable reason is given. **Illness:** a doctor's note must be produced. **Excursion:** excursion form signed by student's teachers. **Other:** appropriate documents. If a student misses a test/quiz without following the aforementioned guidelines then a mark of zero will be awarded.

Assignments will be given for each unit. Performance on assignments will be used to assess the achievement of knowledge and skills.

Late Assignments: all assignments are due at the beginning of the class on the due date, all assignments handed in past the ultimate due date (the last date the assignment will be accepted) will no longer be accepted without a valid reason. If the teacher hands back an assignment to students, after being marked, late assignments will no longer be accepted after unit expectations deadline expired.

Student Evaluation Sheet

Name: _____ Student Number: _____

N/M	Level 1			Level 2			Level 3			Level 4		
0-49	50-52	53-56	57-59	60-62	63-66	67-69	70-72	73-76	77-79	80-88	88-94	94-100
0	1-	1	1+	2-	2	2+	3-	3	3+	4-	4	4+

Assignment/Test/Quiz Description	Overall Mark %	Knowledge/ Understanding	Thinking/ Inquiry	Application	Communication	A) Not submitted B) Missed test/quiz C) Late assignment
1.						
2.						
3.						
4.						
5.						

Independent Study

There will be at least one independent study assigned. The independent study will be based on any of the units outlined for this course. Students are responsible for any material covered by independent study. Thus, tests may cover some of the material from the independent study.

Remember: You always have CLINIC TIME to work on any computer assignments if you fall behind.

Midterm Mark: there will be no midterm examination. The midterm mark is based on the student's cumulative mark at the midterm mark date.

Binder Mark: each student has to keep all lessons, notes, handouts, assignments, etc. well organized in a separate binder with dividers for units and modules. The binder will be marked at least twice: mid term mark and final mark. The mark will be based on how well organized is the binder, completeness of content and assignments. The mark will be include in the learning skills.

Participation Mark: at the end of semester each student will get a participation mark based on the number of absences and lates, class participation, course and class discipline, general attitude towards the course values, equipment and teaching and learning process.

The Final Exam

The content of the final exam will be based on the material covered throughout the full semester. For further information on exam protocol refer to the student agenda book.

Resource Books

Beth Brown, Bruce Presley. *An Introduction to Programming Using Microsoft Visual Basic.Net*, Lawrenceville Press, 2003. ISBN 1-58003-038-6.

Norton, Peter. *Essential Concepts*. McGraw-Hill Ryerson Limited, 1999. ISBN 0-02-804394-4

Rood, H.J. *Logic and Structured Design for Computer Programmers*, Second Edition. Boston: PWS-KENT Publishing Company, 1992. (Chpt. 1.2, 2) ISBN 0-534-92966-4

Tapscott, Don. *Growing up Digital*. McGraw Hill Ryerson Limited, 1998. ISBN 0-07-063361-4

White, Ron. *How Computers Work*. Quebec, Canada: 1997. ISBN 01-56-276546-9

Achievement Charts – ICS20

Student Name Assignment

Categories Knowledge/Understanding Thinking/Inquiry Communication Application

Category	N/M	Level 1			Level 2			Level 3			Level 4		
		0-49	50-52	53-56	57-59	60-62	63-66	67-69	70-72	73-76	77-79	80-88	88-94
Knowledge/Understanding	0	1-	1	1+	2-	2	2+	3-	3	3+	4-	4	4+
<ul style="list-style-type: none"> knowledge of facts, technical terminology, procedures, and standards 		<ul style="list-style-type: none"> demonstrates limited knowledge of facts, technical terminology, procedures, and standards 	<ul style="list-style-type: none"> demonstrates some knowledge of facts, technical terminology, procedures, and standards 	<ul style="list-style-type: none"> demonstrates considerable knowledge of facts, technical terminology, procedures, and standards 	<ul style="list-style-type: none"> demonstrates considerable knowledge of facts, technical terminology, procedures, and standards 	<ul style="list-style-type: none"> demonstrates thorough and insightful understanding of facts, technical terminology, procedures, and standards 							
<ul style="list-style-type: none"> understanding of concepts (e.g., uses of computer operating systems) 		<ul style="list-style-type: none"> demonstrates limited understanding of concepts 	<ul style="list-style-type: none"> demonstrates some understanding of concepts 	<ul style="list-style-type: none"> demonstrates considerable understanding of concepts 	<ul style="list-style-type: none"> demonstrates thorough and insightful understanding of concepts 								
<ul style="list-style-type: none"> understanding of relationships between concepts (e.g., energy conservation and 		<ul style="list-style-type: none"> demonstrates limited understanding of relationships between concepts 	<ul style="list-style-type: none"> demonstrates some understanding of relationships between concepts 	<ul style="list-style-type: none"> demonstrates considerable understanding of relationships between concepts 	<ul style="list-style-type: none"> demonstrates thorough and insightful understanding of relationships between concepts 								

Category	N/M	Level 1			Level 2			Level 3			Level 4		
		0-49	50-52	53-56	57-59	60-62	63-66	67-69	70-72	73-76	77-79	80-88	88-94
Communication	0	1-	1	1+	2-	2	2+	3-	3	3+	4-	4	4+
<ul style="list-style-type: none"> communication of information (e.g., computer & technical specifications) 		<ul style="list-style-type: none"> communicates information with limited clarity 	<ul style="list-style-type: none"> communicates information with moderate clarity 	<ul style="list-style-type: none"> communicates information with considerable clarity 	<ul style="list-style-type: none"> communicates information with a high degree of clarity, and with confidence 								
<ul style="list-style-type: none"> use of language, symbols, and visuals (e.g., computer programming and technical drawing) 		<ul style="list-style-type: none"> uses language, symbols, and visuals with limited accuracy and effectiveness 	<ul style="list-style-type: none"> uses language, symbols, and visuals with some accuracy and effectiveness 	<ul style="list-style-type: none"> uses language, symbols, and visuals with considerable accuracy and effectiveness 	<ul style="list-style-type: none"> uses language, symbols, and visuals with a high degree of accuracy and effectiveness 								
<ul style="list-style-type: none"> communication for different audiences and purposes (e.g., tourism, marketing) 		<ul style="list-style-type: none"> communicates with a limited sense of audience and purpose 	<ul style="list-style-type: none"> communicates with some sense of audience and purpose 	<ul style="list-style-type: none"> communicates with a clear sense of audience and purpose 	<ul style="list-style-type: none"> communicates with a strong sense of audience and purpose 								
<ul style="list-style-type: none"> use of various forms of communication (e.g., presentation software) 		<ul style="list-style-type: none"> demonstrates limited command of the various forms 	<ul style="list-style-type: none"> demonstrates moderate command of the various forms 	<ul style="list-style-type: none"> demonstrates considerable command of the various forms 	<ul style="list-style-type: none"> demonstrates extensive command of the various forms 								

Category	N/M	Level 1			Level 2			Level 3			Level 4		
Application	0-49	50-52	53-56	57-59	60-62	63-66	67-69	70-72	73-76	77-79	80-88	88-94	94-100
	0	1-	1	1+	2-	2	2+	3-	3	3+	4-	4	4+
<ul style="list-style-type: none"> • application of ideas and skills in familiar contexts (e.g., demonstrating good customer service practices) 		<ul style="list-style-type: none"> • applies ideas and skills in familiar contexts with limited effectiveness 			<ul style="list-style-type: none"> • applies ideas and skills in familiar contexts with moderate effectiveness 			<ul style="list-style-type: none"> • applies ideas and skills in familiar contexts with considerable effectiveness 			<ul style="list-style-type: none"> • uses thinking skills with a high degree of effectiveness 		
<ul style="list-style-type: none"> • transfer of concepts, skills, and procedures to new contexts (e.g., applying scientific principles to health care and personal services) 		<ul style="list-style-type: none"> • transfers concepts, skills, and procedures to new contexts with limited effectiveness 			<ul style="list-style-type: none"> • transfers concepts, skills, and procedures to new contexts with moderate effectiveness 			<ul style="list-style-type: none"> • transfers concepts, skills, and procedures to new contexts with considerable effectiveness 			<ul style="list-style-type: none"> • transfers concepts, skills, and procedures to new contexts with a high degree of effectiveness 		
<ul style="list-style-type: none"> • application of procedures, equipment and technology (e.g., use of design instruments, machine and hand tools) 		<ul style="list-style-type: none"> • uses procedures, equipment, and technology safely and correctly only with supervision 			<ul style="list-style-type: none"> • uses procedures, equipment, and technology safely and correctly with some supervision 			<ul style="list-style-type: none"> • uses procedures, equipment, and technology safely and correctly 			<ul style="list-style-type: none"> • demonstrates and promotes the safe and correct use of procedures, equipment, and technology 		
<ul style="list-style-type: none"> • making connections (e.g., between personal experiences and the subject, between subjects, between subjects and the world outside the school) 		<ul style="list-style-type: none"> • makes connections with limited effectiveness 			<ul style="list-style-type: none"> • makes connections with moderate effectiveness 			<ul style="list-style-type: none"> • makes connections with considerable effectiveness 			<ul style="list-style-type: none"> • makes connections with a high degree of effectiveness 		
Category	N/M	Level 1			Level 2			Level 3			Level 4		
Thinking/ Inquiry	0-49	50-52	53-56	57-59	60-62	63-66	67-69	70-72	73-76	77-79	80-88	88-94	94-100
	0	1-	1	1+	2-	2	2+	3-	3	3+	4-	4	4+
<ul style="list-style-type: none"> • thinking skills (e.g., evaluating professional practices and principles) 		<ul style="list-style-type: none"> • uses thinking skills with limited effectiveness 			<ul style="list-style-type: none"> • uses thinking skills with moderate effectiveness 			<ul style="list-style-type: none"> • uses thinking skills with considerable effectiveness 			<ul style="list-style-type: none"> • uses thinking skills with a high degree of effectiveness 		
<ul style="list-style-type: none"> • inquiry/design skills (e.g., identifying the problem; formulating questions; planning; selecting strategies and resources; analysing and interpreting information; forming conclusions) 		<ul style="list-style-type: none"> • applies few of the skills involved in an inquiry/design process 			<ul style="list-style-type: none"> • applies some of the skills involved in an inquiry/design process 			<ul style="list-style-type: none"> • applies most of the skills involved in an inquiry/design process 			<ul style="list-style-type: none"> • applies all or almost all of the skills involved in an inquiry/design process 		