

# **Grade 11 Computer Science (30S)**

A Course for  
Independent Study



---

**GRADE 11**  
**COMPUTER SCIENCE (30S)**

*A Course for Independent Study*

**2010**

Manitoba Education

---

Manitoba Education Cataloguing in Publication Data

Grade 11 computer science (30S) : a course for independent study

ISBN-13: 978-0-7711-4594-0

1. Computer science—Study and teaching (Secondary).
  2. Computer science—Programmed instruction.
  3. Computer science—Study and teaching (Secondary)—Manitoba.
  4. Electronic data processing—Study and teaching (Secondary).
  5. Electronic data processing—Programmed instruction.
  6. Electronic data processing—Study and teaching (Secondary)—Manitoba.
- I. Manitoba. Manitoba Education.  
004

Copyright © 2010, the Government of Manitoba, represented by the Minister of Education.

Manitoba Education  
School Programs Division  
Winnipeg, Manitoba, Canada

Every effort has been made to acknowledge original sources and to comply with copyright law. If cases are identified where this has not been done, please notify Manitoba Education. Errors or omissions will be corrected in a future edition. Sincere thanks to the authors and publishers who allowed their original material to be used.

All images found in this document are copyright protected and should not be extracted, accessed, or reproduced for any purpose other than for their intended educational use in this document.

Any websites referenced in this document are subject to change. Educators are advised to preview and evaluate websites and online resources before recommending them for student use.

## Acknowledgements

Manitoba Education gratefully acknowledges the contributions of the following individuals in the development *Grade 11 Computer Science (30S): A Course for Distance Learning*.

### Course Writer:

Brent Giesbrecht	Course Writer	Green Valley School Hanover S.D.
------------------	---------------	-------------------------------------

### Manitoba Education Staff

#### School Programs Division

Louise Boissonneault	Coordinator	Document Production Services Unit Educational Resources Branch
Lee-Ila Bothe	Coordinator (until March 2010)	Document Production Services Unit Educational Resources Branch
Darryl Gervais	Project Leader	Instruction, Curriculum and Assessment Branch
Lynn Harrison	Desktop Publisher	Document Production Services Unit Educational Resources Branch
Gilles Landry	Project Manager	Development Unit Instruction, Curriculum and Assessment Branch
Grant Moore	Publications Editor	Document Production Services Unit Educational Resources Branch
Lindsay Walker	Desktop Publisher	Document Production Services Unit Educational Resources Branch
Susan Wersch	Coordinator	Distance Learning Unit Instruction, Curriculum and Assessment Branch

## Notes

## Contents

<b>Acknowledgements</b>	<i>iii</i>
<b>Introduction</b>	<i>1</i>
<b>Module 1: Learning to Program</b>	
Lesson 1: Getting Started	<i>3</i>
Lesson 2: Looking towards the End!	<i>7</i>
Lesson 3: Writing Your First Program	<i>13</i>
Lesson 4: Debugging Your Software	<i>23</i>
Assignment 1.1: Programming and Debugging	<i>27</i>
Lesson 5: History of Programming Languages	<i>29</i>
Assignment 1.2: Programming Languages (1)	<i>30</i>
<b>Module 2: Arrays—Beyond the First Dimension</b>	
Lesson 1: Working with Arrays	<i>3</i>
Lesson 2: Sorting Data within an Array	<i>11</i>
Assignment 2.1: Grade Book	<i>19</i>
Assignment 2.2: Grade Tracker	<i>20</i>
Lesson 3: Types of Arrays	<i>21</i>
Assignment 2.3: Character Arrays	<i>29</i>
Assignment 2.4: Random Data	<i>31</i>
Assignment 2.5: File Data 1	<i>33</i>
Assignment 2.6: File Data 2	<i>35</i>
<b>Module 3: Introduction to Windows Programming</b>	
Introduction	<i>3</i>
Lesson 1: A First Windows Program	<i>5</i>
Lesson 2: Adding Interactivity to FirstWinApp	<i>11</i>
Lesson 3: Working with Strings	<i>19</i>
Lesson 4: Working with Numbers	<i>23</i>
Lesson 5: Adding a Menu to the Program	<i>31</i>
Assignment 3.1: Metric Conversion	<i>40</i>
Lesson 6: Documentation	<i>43</i>
Lesson 7: Putting It All Together—A Complete Example	<i>51</i>
Lesson 8: On Your Own	<i>65</i>
Assignment 3.2: Digital Clock	<i>67</i>

Assignment 3.3: String Manipulation	69
Assignment 3.4: GUI Development	75
Assignment 3.5: Online Catalogue	81
Assignment 3.6: A Timing Loop Application	85

#### **Module 4: Independent Studies**

Introduction	3
Lesson 1: Development of Computer Languages	5
Assignment 4.1: Programming Languages (2)	6
Lesson 2: Development of Careers in Computer Science	7
Assignment 4.2: Career Research	8
Lesson 3: Issues in Computing	11
Assignment 4.3: Book Review	12

#### **Module 5: Functions and Data Structures**

Introduction	3
Lesson 1: Functions in C#	5
Lesson 2: Basic One-Dimensional Arrays	9
Lesson 3: Arrays of Controls	13
Assignment 5.1: Controls Array	18
Lesson 4: Objects	21

#### **Module 6: Further Independent Studies**

Introduction	3
Lesson 1: Dialog Research	5
Assignment 6.1: Independent Research	7
Lesson 2: Advanced Controls	9
Assignment 6.2: Advanced Controls #2	15
Assignment 6.3: Scrabble Solver	17
Lesson 3: The ListView Control	21
Assignment 6.4: ListView Control	22
Assignment 6.5: Researching .Net Controls	23
Assignment 6.6: Chequebook	25

#### **Appendix A: Notes to Classroom Teachers, Parents, and Study Partners**

#### **Glossary**

## Grade 11 Computer Science: A Course for Distance Learning (30S)

### Introduction

Welcome to *Grade 11 Computer Science: A Course for Distance Learning*. The purpose of this Introduction is to help you become familiar with the course. It's like the first few days of school, when the teacher tells you about the course and what you are going to learn.

#### What Is Computer Science?

Computers are everywhere. They are in homes, schools, and businesses. Microprocessors are at the core of every computer. Microprocessors are also part of things like video games, cell phones, televisions, and cars. Computer science is about designing and controlling microprocessors. A major part of computer science (and this course) is computer programming.

Computer programming is telling a computer what to do in a language that it can understand. One of those languages is Visual C++, the one that you will be using in this course.

However, computer scientists do more than programming. Some conduct research in areas such as artificial intelligence (trying to understand whether machines can think like people). Some work on computer networks or try to make computers run better and faster. Computer scientists work for many different kinds of organizations, large or small.

#### What Is this Course About?

This is an optional, full-credit computer programming course based on a Microsoft Windows operating system using Microsoft Visual Studio 2010 Express, programming in the C++ and C# language. It is strongly recommended that you already have completed Grade 10 Computer Science before attempting this course.

This course will give you the chance to solve problems, accomplish tasks, and express creativity by using programming techniques. The techniques that you will learn are applicable to other programming languages. This course will enable you to explore and further develop skills in solving problems and prepare you for future study at college or university.

## What Will You Need?

- A computer with the following **minimum** requirements:
  - a 1.6 GHz processor
  - 1024 MB of memory
  - 3 GB of hard disk space
  - **Windows XP** operating system (SP3)
  - a **word processor**
  - an **email** program and **email address** so that you can email your assignments to your tutor/marker
- Access to the Internet and the ability to download and to activate registration keys so you can download the following:
  - Microsoft Visual C++ 2010 Express, available at [www.microsoft.com/express/Downloads/#2010-Visual-CPP](http://www.microsoft.com/express/Downloads/#2010-Visual-CPP).
  - Microsoft Visual C# 2010 Express, available at [www.microsoft.com/express/Downloads/#2010-Visual-CS](http://www.microsoft.com/express/Downloads/#2010-Visual-CS).

(**Note:** Microsoft Visual Studio 2010 Express requires that you register your evaluation copy within **30 days of installation**. Once you have entered your Registration Key, the product is free.)

- Course data files available on the Manitoba Education website at [www.edu.gov.mb.ca/dl/downloads](http://www.edu.gov.mb.ca/dl/downloads).

**Note:** ISO does not provide technical support for hardware-related issues; consult your tutor/marker for software support. If further troubleshooting is required, you may have to consult the software manufacturer or a professional computer technician.

## What If You Need Help?

Taking a distance learning course is different from taking a course in a classroom. Instead of the teacher telling you to complete an exercise, you need to tell yourself to do so. However, there are two people who can help you be successful in your course. They are:



**Your Tutor/Marker:** The first person who can help you is your tutor/marker. Tutor/markers are experienced teachers who tutor distance learning students and mark assignments and exams. When you are having difficulty, be sure to contact your tutor/marker. They are there to help you. You should feel free to contact your tutor/marker at any time during this course. Your tutor/marker's contact information is found on the Record of Progress sent with this course.

**Your Study Partner:** The next person who can help you with your course is your study partner. A study partner is someone you choose who will help you learn. It may be someone who knows something about computer science, but it doesn't have to be. A study partner could be someone else who is taking this course, a teacher, parent, sibling, friend, or anybody else who can help you. Most importantly, a study partner should be someone with whom you feel comfortable and who will help you with the course. Your study partner can help you keep on schedule, check your work, help you make sense of assignments, and give you advice.

### **How Is the Course Organized?**

The course is broken down into six modules. Next to the name of the module is the approximate amount of time that you will need to complete it.

Module 1: Learning to Program (approximately 10 hours)

Module 2: Arrays—Beyond the First Dimension  
(approximately 10 hours)

Module 3: Introduction to Windows Programming  
(approximately 30 hours)

Module 4: Independent Studies (approximately 20 hours)

Module 5: Functions and Data Structures (approximately 20 hours)

Module 6: Further Independent Studies (approximately 20 hours)

Glossary: This contains definitions of important terms.

Each module contains several lessons. A typical lesson includes content that you need to read. Some modules also include learning activities to help you try out what you have read. Since you do not hand in your learning activities, you can compare them with the code from the Data/Examples.

**Assignments:** The course contains a total of 24 assignments. They are worth 70 percent of the final mark of the course. **You will be emailing each assignment to your tutor/marker.** For assignments that include computer code, you must also email the code to your tutor/marker.

### Plagiarism

Plagiarism is the use of another's words or ideas without giving credit where credit is due. Some examples are:

- Downloading material in whole or part from the Internet.
- Copying word-for-word from published or unpublished work.
- Paraphrasing published or unpublished material.

Any of the above noted violations are considered to be serious and may result in the following actions:

- Attending students: The school facilitator will be notified and the student will be required to redo the work.
- Non-attending students: The student will be required to redo the work and a record of the plagiarism will be maintained in his or her student file until the course has been completed.

**Final Exam:** When you have completed Module 6, you will write your final exam. The final exam is worth 30 percent of the final mark of the course. In it, you will write a small computer program that satisfies certain specifications based on what you learned in the course. In order to complete the exam, therefore, you will need access to a computer with Microsoft Visual Studio 2010 Express installed in it.

You will also need two portable storage devices (like DVD-RW discs) on which to save your program (one of these will serve as a backup in case the other one is defective). To save the file onto a DVD-RW, the computer will need to have a DVD burner. You will write the program and save it onto the two blank storage devices. You will then give them to the proctor who will be supervising your exam. The proctor will then mail them to your tutor/marker for evaluation.

Before you finish Module 6, you will need to make arrangements to write your final exam. When you write the final exam, you will be supervised by a proctor. Here is how you apply for an exam:

- If you are attending school, ask your school's Independent Study Option (ISO) Facilitator to add your name to the ISO exam eligibility list. Do this at least three weeks prior to the next scheduled exam week.
- If you are not attending school, check the Examination Request Form for options available to you. The Examination Request Form was mailed to you with this course. Fill in this form and mail or fax it three weeks before you are ready to write the exam. The address is:

ISO Registration  
555 Main Street  
Winkler MB R6W 1C4  
Fax: 204-325-1719

### **How Can You Practise What You've Learned?**

It's really important to practise what you've learned. If you don't, you won't have the confidence or experience when you are working on your assignments or writing your exam. We've included learning activities in some modules to give you the chance to practise what you've learned. You will be working on the learning activities as you complete each lesson and checking your work with the files on the Data/Examples CD. **Do not send your completed learning activities to your tutor/marker.**

### **How Much Time Will You Need?**

Learning computer science through distance learning is a little different than learning it in the classroom. One of the advantages is that you are in charge of how you learn and can choose how quickly you will complete the course. You don't have to wait for your teacher or friends, and you can work as quickly as you want. You can also complete as many lessons at a time as you want. We really want you to succeed. Please read the next few pages to get an idea of how to pace yourself.

On the top of the first page of each lesson, write the date and time when you started and finished it. In this way, you will have a record of the time that you spent on each lesson. As the course progresses, the assignments become more involved and will take

more time. Refer to the time allocations found earlier in this Introduction to make sure you have enough time to complete each module.

Please note that these times are approximations. Depending on many factors, it may take you more or less time to complete the modules. You will be spending a minimum of 110 hours on this course. This includes time spent reading, studying, programming, completing assignments, and writing exams. That means at least 45 minutes daily to complete the course in a regular school year, or at least 90 minutes daily in a semester.

Look at the following three charts and decide which chart best describes the time of year that you want to finish the course. Share your study plans and timelines with your study partner, so that you know that you are on target with your goal of completing this course by a specific target date. Failing to plan is planning to fail. A very important step toward success is being able to plan and manage your time.

### Chart A: Semester 1

Here is a suggested timeline that you can follow if you have registered for this course in September and would like to complete it by January.

Chart A: Semester 1		
Module 1	Assignments 1.1 to 1.2	September 12
Module 2	Assignments 2.1 to 2.6	September 25
Module 3	Assignments 3.1 to 3.6	November 10
Module 4	Assignments 4.1 to 4.3	November 31
Module 5	Assignment 5.1	December 20
Module 6	Assignments 6.1 to 6.6	January 10
Final Exam		January 15

**Chart B: Semester 2**

Here is a suggested timeline that you can follow if you have registered for this course in January and would like to complete it by June.

<b>Chart B: Semester 2</b>		
Module 1	Assignments 1.1 to 1.2	February 10
Module 2	Assignments 2.1 to 2.6	February 20
Module 3	Assignments 3.1 to 3.6	March 31
Module 4	Assignments 4.1 to 4.3	April 20
Module 5	Assignment 5.1	May 10
Module 6	Assignments 6.1 to 6.6	May 31
Final Exam		June 10

**Chart C: Full School Year (Not Semestered)**

Here is a suggested timeline that you can follow if you have registered for this course in September and would like to complete it by June.

<b>Chart C: Full School Year (Not Semestered)</b>		
Module 1	Assignments 1.1 to 1.2	September 20
Module 2	Assignments 2.1 to 2.6	October 10
Module 3	Assignments 3.1 to 3.6	December 20
Module 4	Assignments 4.1 to 4.3	February 28
Module 5	Assignment 5.1	April 10
Module 6	Assignments 6.1 to 6.6	May 15
Final Exam		June 10

Do not wait until the last minute to complete your work, since your tutor/marker may not be available to mark it immediately. Remember, it might take over a week for your work to travel through the mail. It may also take a few weeks for your tutor/marker to mark everything and send your marks to your school.

## Assignment Checklist

Here is a list of all of the assignments that you will be completing in the course.

	Assignment	Mark	Percent of Final Score
	1.1	/60	
	1.2	/65	
<b>Module 1 Total</b>		<b>/125</b>	<b>6%</b>
	2.1	/30	
	2.2	/30	
	2.3	/30	
	2.4	/30	
	2.5	/30	
	2.6	/30	
<b>Module 2 Total</b>		<b>/180</b>	<b>12%</b>
	3.1	/70	
	3.2	/60	
	3.3	/70	
	3.4	/70	
	3.5	/70	
	3.6	/60	
<b>Module 3 Total</b>		<b>/400</b>	<b>20%</b>
	4.1	/30	
	4.2	/50	
	4.3	/50	
<b>Module 4 Total</b>		<b>/130</b>	<b>8%</b>
	5.1	/70	
<b>Module 5 Total</b>		<b>/70</b>	<b>4%</b>
	6.1	/70	
	6.2	/70	
	6.3	/70	
	6.4	/40	
	6.5	/70	
	6.6	/40	
<b>Module 6 Total</b>		<b>/360</b>	<b>20%</b>
<b>Assignment Total for Entire Course</b>			<b>70%</b>

## How Do You Send in Your Assignments?

In this course, you have the choice of either mailing or emailing your assignments.

- Each time that you mail something, you must include the print version of the applicable Cover Sheet (found at the end of this Introduction).
- Each time that you email something, you must include the electronic version of the applicable Cover Sheet (found at <[www.edu.gov.mb.ca/k12/dl/downloads/index.html](http://www.edu.gov.mb.ca/k12/dl/downloads/index.html)>).

Complete the information at the top of the Cover Sheet before mailing or emailing it along with your assignments.

## Mailing Your Assignments

If you choose to mail your completed assignments, please photocopy all of the materials first so that you will have a copy in case your package goes missing. You will need to place the applicable module Cover Sheet and assignments in an envelope and address it to

ISO Tutor/Marker  
555 Main Street  
Winkler MB R6W 1C4

Your tutor/marker will mark your work and return it to you by mail.

## Emailing Your Assignments

If you choose to email your assignments, make sure you have saved copies of them before you send them. That way, you can refer to your assignments when you discuss them with your tutor/marker.

To email your completed assignments, you will first need to do one of the following:

- **If you are attending school**, please ask your ISO school facilitator (the person who signed your ISO Registration/ Admission Form) for permission to email your assignments and to determine your school's procedure for emailing assignments/unsupervised tests. Contact your tutor/marker to confirm that the course material can be marked electronically.

- **If you are not attending school**, please obtain permission directly from your tutor/marker to submit your assignments electronically.

### ***How to Submit Your Work (files must not exceed 5 MB)***

Please submit your work in the file types shown below:

1. Written work: Microsoft Word files (doc) or RTF files
2. Spreadsheets: Microsoft Excel files (xls)
3. Pictures and graphics: JPEG or GIF files
4. Scanned file: PDF files (save multiple pages as one file)
5. Audio recordings: WAV files
6. Video recordings: WMV files

### ***How To Send Your Email***

1. Use the following format to compose your email.

To:	distance.learning@gov.mb.ca
cc:	(your ISO school facilitators' email address if you attend school)
Subject:	[My Name] Grade 11 Computer Science
Attach:	Assignment 1.1.doc
Message:	Assignment 1.1, 1.2, and 1.3
	Tutor/marker: _____
	School: _____

2. Attach your files (files must not exceed 5 MB).
3. Email your assignments to <distance.learning@gov.mb.ca>. Do **not** email your assignments directly to your tutor/marker. Emails sent directly to your tutor/marker will be returned unread.

Your tutor/marker will electronically mark your work and email it back to you.

## Guide Graphics

Guide graphics are included throughout the course to help you identify specific tasks you need to complete. Here is an explanation for each of them:



**Learning Activity:** Complete this to practise what you have learned.



**Assignment:** This icon indicates there is an assignment you will complete for marks that needs to be either mailed or emailed to your tutor/marker.



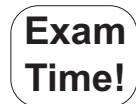
**Note:** This icon signifies important and significant material.



**Remember:** This icon reminds you of important information.



**Outcomes:** This icon indicates what you will learn.



**Exam Time:** This icon reminds you that it is time to make arrangements to write your final examination.

## Notes



---

**GRADE 11**  
**COMPUTER SCIENCE (30S)**

*Module 1*  
*Learning to Program*

---



## Lesson 1

### Getting Started

#### Introduction

You are about to embark on a journey that potentially will be one of your most rewarding paths of study. This course continues from the Grade 10 Computer Science course and introduces you to additional programming concepts. One of the topics that we will cover in this course is the development of a Windows-level programming environment that allows us to write programs with menus, dialog boxes, pens, brushes, bitmaps, cursors, icons, and all those other fun features that we don't have in the console programs you wrote in Grade 10 Computer Science.

As you probably discovered in the Grade 10 course, computer programming brings with it many joys. Your computer is now an extension of your brain. If you want to figure out something, you can write a program to do it for you. Imagine, for virtually anything you think, you can write a program to do the task. When you take the leap into Windows programming, you also have the opportunity to express your artistic side by developing user graphical interfaces.

You may also remember that computer programming also brings with it many frustrations. The frustrations stem from the fact that the computer will execute your program exactly as you asked it to. So, if you did not type the program perfectly, it will not be perfect. Avoid frustration; check your work carefully.

#### Outcomes

When you have completed this lesson, you will be able to:

- Identify your computer's capabilities
- Ensure that your software is correctly installed

#### Overview

Following is a list of topics covered in this lesson:

- Programming requirements
- Where we are headed...



## Programming Requirements

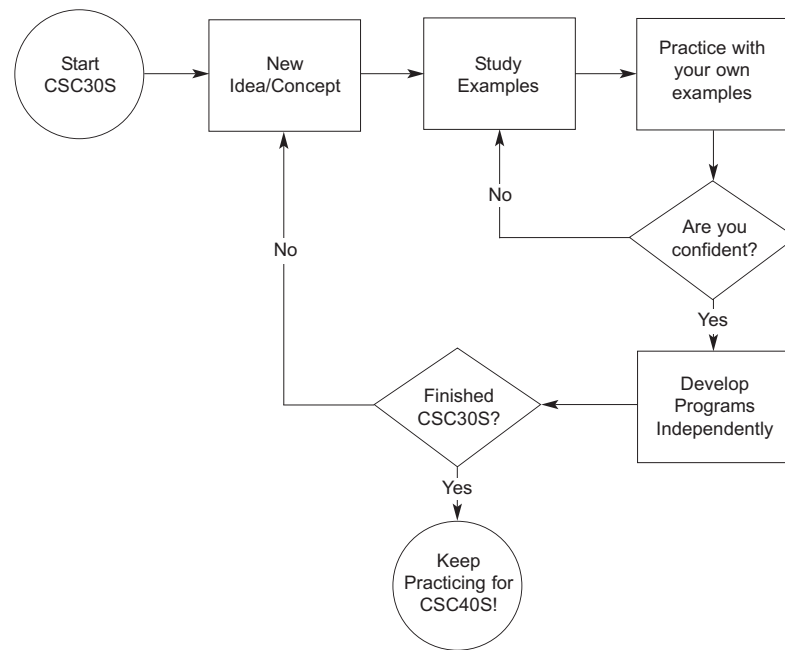
To continue in this course you must ensure that you have the appropriate computing equipment available. You should have (as a minimum):

- Windows computer with Pentium class processor or higher, 64MB of memory, 5+GB hard disk
- Monitor Resolution:
  - If your monitor is less than 17", use a window resolution of 800x600.
  - If your monitor is 17" or greater, consider a resolution of at least 1024x768.
- Windows operating system.
- Microsoft Visual Studio 2010 Express or Microsoft C# installed on your computer

## About Computer Programming

This course is structured around a process of learning by practice. As you proceed through the course material, you will be continually exposed to new ideas. These new ideas will be illustrated through a variety of examples after which you will be expected to demonstrate your understanding through programming some examples on your own.

Finally, to determine your ability to use these new concepts, you will be required to complete ever larger programming projects. These projects will often be comprehensive in nature. This means that, as you learn more about programming, your programs will reflect all the skills you have developed at that point in time.



### Summary

At this point you should be confident that all the resources you require to complete this course are in place. If you have any concerns about the software installation or about the computer with which you are working, please contact your tutor/marker. You must be sure you are ready to undertake this course and there will be no surprises later on.

### End of Lesson

Before you move towards Windows programming, you still have to learn a lot of programming concepts. In order to do this, this course will focus on the console application mode a while longer. The jump to Windows applications will happen soon enough. To give you a taste of where this course is headed, the next lesson consists of a series of demo programs for you to explore.

Just to make sure you understand, you will be asked to study the next two sections as review from last year. At that time you became familiar with the Microsoft Visual Studio programming environment. However, if you are like most students, you may already have forgotten some of the specific things you learned in Grade 10 Computer Science. So you will briefly review project development and debugging.

## Notes





Released 2010



Printed in Canada  
Imprimé au Canada