



## Board/Authority Authorized Course Framework Template

<b>School District/Independent School Authority Name:</b> School District #33 Chilliwack	<b>School District/Independent School Authority Number (e.g. SD43, Authority #432):</b> SD33
<b>Developed by:</b> Darren Watt	<b>Date Developed:</b> <b>Revised 2024</b> , (based on courses that have evolved and continually improved and taught under different names since 1996)
<b>School Name:</b> Sardis Secondary	<b>Principal's Name:</b> Schramm
<b>Superintendent Approval Date (for School Districts only):</b>	<b>Superintendent Signature (for School Districts only):</b>
<b>Board/Authority Approval Date:</b>	<b>Board/Authority Chair Signature:</b>
<b>Course Name:</b> Business Information Management 12	<b>Grade Level of Course:</b> 12
<b>Number of Course Credits:</b> 4	<b>Number of Hours of Instruction:</b> 120

**Board/Authority Prerequisite(s):**

No pre-requisite courses.

**Special Training, Facilities or Equipment Required:**

Microsoft Office Software and computer.

**Course Synopsis:**

Use advanced knowledge of application software tools to design an integrated business application to accomplish tracking records or financial transactions and utilize structured programming coding to compliment application software where there are limits to the advanced tools.

**Goals and Rationale:**

Goes beyond traditional uses of application software into advanced areas and incorporates programming language.

**Aboriginal Worldviews and Perspectives:**

This material is applicable to emerging Indigenous business opportunities and is free from bias and the perspective of privilege. Projects can be customized to aboriginal worldviews and perspectives.

### BIG IDEAS

Business decisions and the continual tracking of the flow of information can be made through **advanced knowledge of application software tools.**

**Design an integrated application** using application software to accomplish tracking records or financial numbers.

**Utilize custom programming languages** to compliment application software where there are limits to even the utilization of advanced tools.

### Learning Standards

Curricular Competencies	Content
<p><i>Students are expected to do the following:</i></p> <p><b>Applied Design</b></p> <ul style="list-style-type: none"> <li>Choose an appropriate form, scale, and level of detail for communicating outcomes in a clear and concise manner.</li> <li>Use reporting designs consistent with <b>established business reporting.</b></li> <li>Use custom designs consistent with the <b>unique requests of individual business or users.</b></li> </ul> <p><b>Applied Skills</b></p> <ul style="list-style-type: none"> <li><b>Take inventory of your skills</b> and determine the level of complexity that you can add to your <b>custom-made application software.</b></li> <li>Incorporate built in easy and <b>advanced options</b> as appropriate for the intended use.</li> <li><b>Research the feasibility and optional features</b> outside of your current knowledge.</li> <li>Peruse <b>templates</b> and online examples for ideas on custom applications using pre-made solutions where appropriate.</li> </ul>	<p><i>Students are expected to know the following:</i></p> <ul style="list-style-type: none"> <li>Advanced <b>database design</b> and application:                             <ul style="list-style-type: none"> <li>database creation, modification, and navigation,</li> <li><b>tables, queries,</b> forms, and report design,</li> <li>variety of field types including <b>calculated fields,</b></li> <li>role of type of <b>data relationships</b> and <b>primary keys,</b></li> <li>delete, append, update <b>queries,</b></li> <li>grouping, sorting, and parameters in <b>forms and reports,</b></li> <li><b>form controls and report options,</b></li> <li>functions, action queries, custom buttons, dialogue boxes,</li> <li><b>importing and exporting various data,</b></li> <li><b>pop up calendar</b> for data entry,</li> <li><b>drop lists</b> for data entry, and</li> </ul> </li> </ul>

## Applied Technologies

- Use application templates as a starting point if useful.
- Consider writing **Microsoft Office Specialist exams** to solidify your strengths in spreadsheets and databases and consider other adjunct Microsoft products that may be incorporated or more suitable.
- **Incorporate structured programming language** as an adjunct to application software for advanced features that can only be coded from line coding.
  - database for **school data** (student records, book sign out, course selection), **rental store** (sign in, out calculate costs), and **dentist office** (appointment/invoicing) creation.
- **Advanced database computer programming:**
  - programming language eg. **Visual Basic for Applications**,
  - coding for **message boxes** and **input boxes** attached to a linked database,
  - coding for **locking** various fields,
  - coding for **password access**,
  - coding for **variable declaration**,
  - coding for **conditional statements**,
  - coding for **repetitive loops**, (do until and do while),
  - coding to activate action queries, and
  - coding for **warnings**.
- **Advanced spreadsheet design** and application:
  - advanced **functions** for mortgages and annuities,
  - creation and analysis of mortgage tables.
  - **implications of changes in interest** or mortgage terms,
  - **implications of risk factors on net present value** of investments,
  - incorporation of **gross debt ratio** and factors such as insurance, down payment, property taxes into calculated monthly payments,
  - comparing **lease versus purchase** options numerically
  - **various pensions** such as government, company, and self-financed RRSP calculations
  - calculations of net present value of investment options considering **risk assessment**

- advanced computer **programming integration:**
  - programming language eg. **Visual Basic for Applications.**
  - coding for message boxes and input boxes attached to a linked spreadsheet,
  - coding for selecting ranges and cell options boxes,
  - coding for password access boxes,
  - coding for variable declaration,
  - coding for conditional statements,
  - coding for repetitive loops (do until and do while),
  - coding for date calculation and insertion,
  - coding for incrementing variable,
  - coding for warnings,
  - coding for **offset method of selection,**
  - coding for **selecting and deleting selections,**
  - coding to **use variables in calculations,**
  - coding to execute **sub routines based on conditional statements and user input,** and
  - **debugging errors** in code

## Big Ideas – Elaborations

- **advanced knowledge of application software tools:** many advanced tools expand the use of commonly used application software for advanced applications.
- **design an integrated application:** beyond creating one-time documents, an integrated application acts as a system within the business flows on a continual basis.
- **utilize custom programming languages:** beyond even the advanced features, coding can be used adjunct to application software to expand its use.

## Curricular Competencies – Elaborations

- **established business reporting:** there are common financial document conventions to follow and practical data presentation examples available to look at in the planning phase.
- **unique requests of individual business or users:** customizing application software goes beyond basic uses and acts as a custom-made solution short of a totally coded software solution.
- **take inventory of your skills custom-made application software:** recognize your limits and access to additional expert assistance.
- **advanced options:** familiarize yourself with the advanced features of the software.
- **research the feasibility and optional features:** customized solutions may be available through online forums.
- **Microsoft Office Specialist exams:** standardized level of designer.
- **Incorporate structured programming language:** Microsoft Basic for Applications coding can integrate with spreadsheet and database software for a more complete business solution.

## Content – Elaborations

- **database design:** application design based on user needs.
- **tables, queries:** basic elements to store and retrieve data.
- **calculated fields:** using stored data in calculations.
- **data relationships:** eg. One to many, one to one.
- **primary keys:** fundamental linking relationship between fields.
- **queries:** extraction of data based on criteria.
- **forms and reports:** presentation of data for user interface or viewing.
- **form controls and report options:** user flexibility to modify.
- **importing and exporting various data** means by which to bring in or extract data for other programs.

## Content – Elaborations

- **pop up calendar:** alternative to manual entry.
  - **drop lists:** alternative to manual entry and restrictive.
  - **school data:** eg. Student number, name, address, course selections, books signed out.
  - **rental store:** eg. Customer information, product loaned, date of loan, calculation of days loaned, calculation of fees and taxes.
  - **dentist office:** eg. Patient information, appointment dates, services performed, calculation of fees and taxes.
  - **programming:** line by line programming code like many other programming languages.
  - **message boxes:** pop up boxes to display messages in certain events.
  - **input boxes:** pop up boxes to ask user for input.
  - **password access:** restrict access to subroutines based on a password.
  - **variable declaration:** universal programming concept of holding data in various types of formats.
  - **conditional statements:** proceeding to additional code based on true/false or other criteria.
  - **repetitive loops:** repeating lines of code based on criteria.
  - **warnings:** to prevent unintended consequences, user must confirm further action.
  - **spreadsheet design:** creation of data in tables and utilizing various formulas, formatting, and summarizing, sorting and other options.
  - **functions:** common built-in tools for complex calculations.
  - **implications of changes in interest:** manipulation of numbers and impact on payments, total interest etc.
  - **implications of risk factors on net present value:** incorporating risk assessment into numerical number.
  - **gross debt ratio:** used by lending associations to determine eligibility.
  - **various pensions:** earnings from government, savings, or employer contributions after retirement.
  - **risk assessment:** a percentage of risk attributed to investment.
- 
- **programming integration:** line by line programming that integrates with a software application for example Access or Excel
  - **Visual Basic for Applications:** commonly used language like programming in Microsoft Environment and used in networks, games etc.
  - **offset method of selection:** automatically select cells based on code.
  - **selecting and deleting selections:** automatically deleting data based on code.
  - **use variables in calculations:** use stored amounts to make calculations.
  - **sub routines:** mini programs within the main program.
  - **conditional statements and user input:** perform task based on a True/False value or what a user is asked.
  - **debugging errors:** noting the type of error and making the necessary corrections

## Content – Elaborations

### **Recommended Instructional Components:**

Can be delivered by self-paced computer lessons and accompanied by knowledgeable instructor for one-on-one lessons for additional clarity.

### **Recommended Assessment Components: Ensure alignment with the [Principles of Quality Assessment](#)**

In line with flexible measuring of outcomes consistent with SD33

### **Learning Resources:**

*BeginCourse: Intro to Advanced courseware* and accompanying Microsoft Office Specialty Exam prep materials, authored by Darren Watt

### **Additional Information:**