# VB

# OBJECT ORIENTED PROGRAMMING

# What is Object Oriented Programming

# WHAT IS OOP?

- Object oriented programming is a design for programming.
- OOP allows us to make one template that creates similar objects
- OOP was created back in the 1950's, but wasn't popularized until the late 70's early 80's

INSTITUTE OF
TECHNOLOGY
DEVELOPMENT
OF CANADA

475 GRANVILLE STREET, VANCOUVER, BC, V6C 1T1
PHONE:      +1(604)558-8727, +1(604)409-8200
TOLL FREE: +1(888) 880-4410
FAX:        +1(888) 881-6545
WEB:        WWW.ITDCANADA.CA
EMAIL:      STUDYING@ITDCANADA.CA

# WHY OOP

# Why OOP?

- Objects are made from a reusable template that allows us to organize our code
- OOP allows us to store objects in memory and use them in unique ways
- Instead of procedural programming, which breaks down tasks to smaller portions such as functions, We use building blocks to write code
- These building blocks are called objects
- Objects are created from templates we make
- The templates are called classes, which we will talk about later

INSTITUTE OF
TECHNOLOGY
DEVELOPMENT
OF CANADA

475 GRANVILLE STREET, VANCOUVER, BC, V6C 1T1
PHONE:      +1(604)558-8727, +1(604)409-8200
TOLL FREE: +1(888) 880-4410
FAX:         +1(888) 881-6545
WEB:        WWW.ITDCANADA.CA
EMAIL:      STUDYING@ITDCANADA.CA

# WHY OOP

- Can you use objects made in one language in another language?

- Yes!

- By using a DLL(Dynamic Link Library) you can make a class in Java and use it in Python or any OOP language into another.

- Windows Forms has objects that are both used in VB and C#.

- These are objects that are written C# but can be used in both languages.

- Database objects like from SQL Client can be used in many different languages

# Real World Objects

INSTITUTE OF
TECHNOLOGY
DEVELOPMENT
OF CANADA

INSTITUTE OF
TECHNOLOGY
DEVELOPMENT
OF CANADA

475 GRANVILLE STREET, VANCOUVER, BC, V6C 1T1
PHONE:     +1(604)558-8727, +1(604)409-8200
TOLL FREE: +1(888) 880-4410
FAX:        +1(888) 881-6545
WEB:       WWW.ITDCANADA.CA
EMAIL:     STUDYING@ITDCANADA.CA

# Object in the real world

- Objects in the real world have two things that define them

    1. They have properties
    2. They have functions

- Lets look at a phone

# Properties - Phone

- A property for a phone can be things like:
  - Color
  - Brand
  - RAM
  - Camera
  - Apps

INSTITUTE OF
TECHNOLOGY
DEVELOPMENT
OF CANADA

475 GRANVILLE STREET, VANCOUVER, BC, V6C 1T1
PHONE:      +1(604)558-8727, +1(604)409-8200
TOLL FREE: +1(888) 880-4410
FAX:         +1(888) 881-6545
WEB:        WWW.ITDCANADA.CA
EMAIL:      STUDYING@ITDCANADA.CA

# FUNCTIONS

- Functions for a phone can be things like:
  - Texting
  - Calling
  - Connect to the Internet
  - Harvest your data
  - Take picture
  - Charge Battery

INSTITUTE OF
TECHNOLOGY
DEVELOPMENT
OF CANADA

475 GRANVILLE STREET, VANCOUVER, BC, V6C 1T1
PHONE:        +1(604)558-8727, +1(604)409-8200
TOLL FREE: +1(888) 880-4410
FAX:            +1(888) 881-6545
WEB:            WWW.ITDCANADA.CA
EMAIL:        STUDYING@ITDCANADA.CA

# OBJECT IN THE REAL WORLD

- Tell me About a restaurant…

- What are its Properties

- What are its Functions

INSTITUTE OF
TECHNOLOGY
DEVELOPMENT
OF CANADA

475 GRANVILLE STREET, VANCOUVER, BC, V6C 1T1
PHONE:     +1(604)558-8727, +1(604)409-8200
TOLL FREE: +1(888) 880-4410
FAX:       +1(888) 881-6545
WEB:       WWW.ITDCANADA.CA
EMAIL:     STUDYING@ITDCANADA.CA

# OBJECT IN THE REAL WORLD

- Tell me About a restaurant…

- Properties: chairs, tables, cash register, menus, building, stove, lights, oven, equipment, website

- Functions: taking the orders, preparing the meal, deliver the meal, the payment process, Server customers, online order taker, food prep, create waste, create employment, buy materials, pay tax

# OBJECT IN THE REAL WORLD

- Tell me About a Dog:

- Properties: Tail, four legs, Nose, Eyes, Fur, Ears, Teeth, Body,

- Functions: Bark, sit, jump, wag tail, run, sleep, eat, play.

INSTITUTE OF
TECHNOLOGY
DEVELOPMENT
OF CANADA

475 GRANVILLE STREET, VANCOUVER, BC, V6C 1T1
PHONE:      +1(604)558-8727, +1(604)409-8200
TOLL FREE: +1(888) 880-4410
FAX:          +1(888) 881-6545
WEB:          WWW.ITDCANADA.CA
EMAIL:        STUDYING@ITDCANADA.CA

# DEPENDENCY PRINCIPAL

- Dependency Principal

# Classes Instantiation And objects

# Classes and Instantiation

- In OOP we can use classes as templates
- To make an object "come to life" it is called instantiation
- A class is abstract, intangible and has no real substance
- An object is real, physical and can be worked with

INSTITUTE OF
TECHNOLOGY
DEVELOPMENT
OF CANADA

475 GRANVILLE STREET, VANCOUVER, BC, V6C 1T1
PHONE:        +1(604)558-8727, +1(604)409-8200
TOLL FREE: +1(888) 880-4410
FAX:          +1(888) 881-6545
WEB:         WWW.ITDCANADA.CA
EMAIL:       STUDYING@ITDCANADA.CA

# CLASSES

# Classes

- Classes are templates for objects
- Classes are like a blueprint for a car. A car has specific properties and functions defined in the blueprint but the template is not the car itself.
- Classes are where we define properties and functions

# MAKING A CLASS

INSTITUTE OF
TECHNOLOGY
DEVELOPMENT
OF CANADA

475 GRANVILLE STREET, VANCOUVER, BC, V6C 1T1
PHONE:      +1(604)558-8727, +1(604)409-8200
TOLL FREE: +1(888) 880-4410
FAX:          +1(888) 881-6545
WEB:          WWW.ITDCANADA.CA
EMAIL:       STUDYING@ITDCANADA.CA

# OOP

- Objects are held in a class
- Objects have two defining features:
    1. They have properties (Data Member Properties)
    2. They can do things (Member Function Methods)
- Data members are the variables for the class
- Member functions are the method for the class
- Both Data members and member functions have access modifiers which define the accessibility of the property/function
    - Public
    - Private

# Static members

- If I make a variable in my class and change the variables value in the My first Object it will not affect the other objects

- The only time this is not true is when you are using static members

- Static members go out of scope

# CONSTRUCTOR AND DESTRUCTOR

- We have two special functions
  1. Constructor(initializer)
  2. Destructor(finalizer)

- They run automatically

- We don't call these functions, they run at the time the object is created or destroyed

- We don't always need a destructor, but you always need a constructor

- As sad as it sounds you always have a day of your birth, but not everyone has a funeral

- In different languages they have different names

INSTITUTE OF
TECHNOLOGY
DEVELOPMENT
OF CANADA

475 GRANVILLE STREET, VANCOUVER, BC, V6C 1T1
PHONE:      +1(604)558-8727, +1(604)409-8200
TOLL FREE: +1(888) 880-4410
FAX:         +1(888) 881-6545
WEB:        WWW.ITDCANADA.CA
EMAIL:      STUDYING@ITDCANADA.CA

# INSTANTIATION

Institute of
Technology
Development
of Canada

475 Granville Street, Vancouver, BC, V6C 1T1
Phone:      +1(604)558-8727, +1(604)409-8200
Toll Free: +1(888) 880-4410
Fax:          +1(888) 881-6545
Web:         WWW.ITDCANADA.CA
Email:      STUDYING@ITDCANADA.CA

# Instantiation

- Instantiation is the process of making or creating an object
- When you have finished making the blueprint of a car, you can make or *instantiate* the car in the factory
- When we instantiate or "create" and object, memory is allocated in RAM and all the data for the object is stored in memory
- Instantiation will allow our properties to be defined
- This would be like when you pick a color for a car, in the factory they make that car the color you picked

# OOP Implementations

- ## Step 1:
  - Define your data members

- ## Step 2:
  - Default constructor 100% necessary
  - Optional: overloaded constructors, these are similar to default but they take arguments
  - Copy constructor takes one argument from its own type; this is for when you create an object from already existing objects(you build an object from an existing one)

- ## Step 3:
  - Develop your access functions (Sets and gets)

- ## Sept 4:
  - Develop your Utility functions, sometimes called helper functions, that do things to data members

- ## Step 5:
  - Instantiate

# OOP

- Assignment

- Make a Class Called My Date

- With data members  Day Month Year

- Methods CheckDay(), CheckMonth(), IsLeap()

INSTITUTE OF
TECHNOLOGY
DEVELOPMENT
OF CANADA

475 GRANVILLE STREET, VANCOUVER, BC, V6C 1T1
PHONE:      +1(604)558-8727, +1(604)409-8200
TOLL FREE: +1(888) 880-4410
FAX:          +1(888) 881-6545
WEB:         WWW.ITDCANADA.CA
EMAIL:      STUDYING@ITDCANADA.CA

# OOP

- Assignment

- Make a Class Called My Date

- With data members  Day Month Year

- Methods CheckDay(), CheckMonth(), CheckYear(), IsLeap()

# Four Fundamentals Of OOP

INSTITUTE OF
TECHNOLOGY
DEVELOPMENT
OF CANADA

475 GRANVILLE STREET, VANCOUVER, BC, V6C 1T1
PHONE:      +1(604)558-8727, +1(604)409-8200
TOLL FREE: +1(888) 880-4410
FAX:          +1(888) 881-6545
WEB:        WWW.ITDCANADA.CA
EMAIL:      STUDYING@ITDCANADA.CA

# Four Fundamentals of OOP

- The Four Fundamentals of OOP are:
  1. Abstraction
  2. Inheritance
  3. Encapsulation
  4. Polymorphism

# Encapsulation

# ENCAPSULATION

- Encapsulation is the containment of our data members and member functions in a class.

- We use encapsulation to make our software more manageable, reduce bugs, and organize our code

- Think about medicine or fish oil. They are put into capsules to prevent the compounds from being mixed together. This is what encapsulation does.
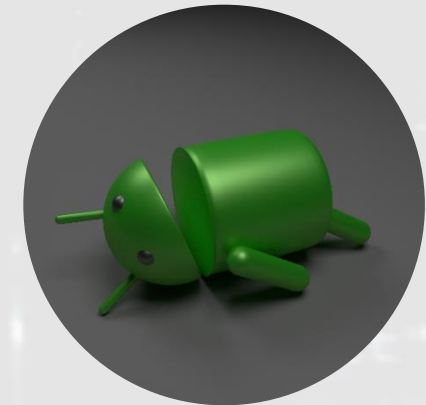
- Encapsulation is a product of using classes

INSTITUTE OF
TECHNOLOGY
DEVELOPMENT
OF CANADA

475 GRANVILLE STREET, VANCOUVER, BC, V6C 1T1
PHONE:      +1(604)558-8727, +1(604)409-8200
TOLL FREE: +1(888) 880-4410
FAX:          +1(888) 881-6545
WEB:         WWW.ITDCANADA.CA
EMAIL:      STUDYING@ITDCANADA.CA

# Encapsulation Vs Data Hiding

- Encapsulation it is commonly thought to be the process of "Data or Information Hiding", but that is a common misconception. Data hiding is a feature of encapsulation but you can encapsulate without hiding data

- Think about the fish oil capsule. You can see the fish oil, so it is not hidden but it is still encapsulated

INSTITUTE OF
TECHNOLOGY
DEVELOPMENT
OF CANADA

475 GRANVILLE STREET, VANCOUVER, BC, V6C 1T1
PHONE:      +1(604)558-8727, +1(604)409-8200
TOLL FREE: +1(888) 880-4410
FAX:          +1(888) 881-6545
WEB:         WWW.ITDCANADA.CA
EMAIL:       STUDYING@ITDCANADA.CA

# DATA HIDING

- Data hiding is where we set the level of accessibility an objects properties and functions have.

- You as a human have an age, I can't change your age or even know your age without the proper security clearance. This is an encapsulated property you have that is hidden from the world.

- A game enemy has a numerical location. As the player you cannot see the enemy numerical location. This data is hidden.

INSTITUTE OF
TECHNOLOGY
DEVELOPMENT
OF CANADA

475 GRANVILLE STREET, VANCOUVER, BC, V6C 1T1
PHONE:      +1(604)558-8727, +1(604)409-8200
TOLL FREE: +1(888) 880-4410
FAX:         +1(888) 881-6545
WEB:        WWW.ITDCANADA.CA
EMAIL:      STUDYING@ITDCANADA.CA

# TAKE AWAYS FROM ENCAPSULATION

- Encapsulation keeps data member and member functions together

- Encapsulation is not data hiding

- Data hiding is a feature of encapsulation

- Encapsulation helps organize code to prevent bugs and increase manageability

INSTITUTE OF
TECHNOLOGY
DEVELOPMENT
OF CANADA

475 GRANVILLE STREET, VANCOUVER, BC, V6C 1T1
PHONE:      +1(604)558-8727, +1(604)409-8200
TOLL FREE: +1(888) 880-4410
FAX:         +1(888) 881-6545
WEB:        WWW.ITDCANADA.CA
EMAIL:      STUDYING@ITDCANADA.CA

# Abstraction

INSTITUTE OF
TECHNOLOGY
DEVELOPMENT
OF CANADA

475 GRANVILLE STREET, VANCOUVER, BC, V6C 1T1
PHONE:      +1(604)558-8727, +1(604)409-8200
TOLL FREE: +1(888) 880-4410
FAX:        +1(888) 881-6545
WEB:        WWW.ITDCANADA.CA
EMAIL:      STUDYING@ITDCANADA.CA

# Abstraction

- Abstraction is the process of abstracting processes from users.

- Abstraction hides processes and implementations

- Abstraction keeps what isn't important to the users abstracted away from them.

- It allows users to use the interface vs having to understand how the interface works.

- You don't need to know how a combustion engine works to drive a car. The engine is abstracted away from the driver.

# Abstraction

- With abstraction its better to have a simple interface with high abstraction vs a complex interface with minimal abstraction.

- Google search has a lot going on behind the scenes. But we only see a search bar and a few buttons

Institute of
Technology
Development
of Canada

475 Granville Street, Vancouver, BC, V6C 1T1
Phone:      +1(604)558-8727, +1(604)409-8200
Toll Free: +1(888) 880-4410
Fax:        +1(888) 881-6545
Web:        WWW.ITDCANADA.CA
Email:      STUDYING@ITDCANADA.CA

# Abstraction

- Abstraction is a design concept and a process of implementation

- There are abstract classes that we can use that can not be instantiated and are used as high level classes which many other classes are inherited from

# Take aways from Abstraction

- Abstraction hides implementation from the end user

- Abstraction is different than Data hiding

# ABSTRACTION VS ENCAPSULATION

- http://www.tonymarston.co.uk/php-mysql/abstraction.txt

# Setters and Getters

# GETTERS AND SETTERS

- When we encapsulate and make data members private, the objects data members can't be accessed from the outside of the object.

- By using getters and setters the object can have its data modified. Think about Textbox1.text. The .text is a getter and setter that allows us to change or get the text value of a Textbox.

- Getters get data from the object

- Setters set data in the object

- Getters are also know as Accessors

- Setters are also know as Mutators

# Getters and Setters

- Getters and Setters are a feature of abstraction and encapsulation. (depending who you ask)

- When we encapsulate and make data members private, the objects data members can't be accessed from the outside of the object.

- We could make the data members public but this reduces data integrity

- Getters and setters allow us to get information from the user and process the data without expectations from the user

- Let say you want to get some data from a user about their age and they input their age as -1  or 500 that could cause issues. When we use getters and setters we can process the data based on the parameters we define

# GETTERS AND SETTERS

- Should we use Getters and Setters?

- Getters and setters technically break encapsulation.

- Its like taking a tiny needle and extracting some of the fish oil from our fish oil capsule.

- Generally speaking its good to use them, but be mindful with them

- Don't make give every data member a getter and setter. This can cause vulnerabilities in you objects.

- Getters can expose the data you want to hide

- Setters can have unwanted data injections into them

- Do the logic for your user, make your functions work for them vs having them doing the work for you