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 in C#

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 Oriented Programming

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 Oriented Programming

- Topics Covered:
  1. What is OOP
  2. Objects In the real World
  3. Four Fundamentals of OOP
  4. Classes, Objects, Instantiation
  5. Class Components
  6. Object Composition

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

What is Object Oriented Programming

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

What is OOP?

- Today we are going to be starting with Object-Oriented Programming, also known as OOP. OOP is an incredibly powerful and widely-used programming paradigm that is utilized in a vast number of programming languages such as Java, Python, C++, and many others.

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 History

- Object-Oriented Programming (OOP) has its roots in the 1960s, when computer scientists began exploring new ways to organize and structure code. A group of researchers at MIT developed the Lisp programming language, which was designed to support modular, reusable code. This laid the foundation for OOP, which emerged in the 1970s as a way to represent complex systems using objects that had unique properties and behaviors. One of the pioneers in this field was Alan Kay, who coined the term "object-oriented programming" and believed that this approach had the potential to revolutionize software development.

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 History?

- Kay's ideas and contributions helped pave the way for the development of modern OOP languages like C++, Java, and Python, which have become some of the most widely used programming languages in the world. OOP has enabled developers to create more efficient, adaptable, and scalable software systems, which has had a profound impact on industries ranging from finance to healthcare to entertainment. The continued evolution of OOP is sure to shape the future of software development for years to come.

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

What is OOP?

- Object-Oriented Programming (OOP) is a way of writing code that focuses on objects. An object is like a container that holds information and actions. A class is like a blueprint for an object, it tells us what the object should look like and what it can do. You can think of it like a cookie cutter that shapes the dough into a specific cookie. With OOP, we can create neat, organized code that is made up of simple building blocks that we can use over and over again to create more complex programs.

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 Benifits?

- One of the key benefits of OOP is its ability to organize code in a way that is more modular, reusable, and easier to understand. By breaking down code into classes and objects, we can create code that is more flexible and adaptable to changes, which is especially important in today's rapidly evolving digital landscape.

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?

- When programmers write code, they can use different methods to solve problems. OOP is like building with Lego blocks - programmers can create objects with specific characteristics and behaviors, then connect them to make a program. This method has some advantages over other methods. For example, OOP makes it easier to keep code organized and easier to work with. Think of it like a toolbox - each object is like a tool, and they can be organized into a toolbox that makes it easy to find the right tool for the job.

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?

- Overall, OOP is a powerful tool that allows programmers to create organized, flexible, and adaptable code. By using OOP, programmers can build more efficient, scalable, and maintainable software systems.

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

Real World Objects

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

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

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 Dog…
- 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 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

Main Concepts

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

- Here, I will provide an overview of the fundamental building blocks of Object-Oriented Programming (OOP), including classes, objects, constructors, destructors, properties, methods, inheritance, polymorphism, encapsulation, and abstraction. While we won't be diving deep into each of these concepts right now, we will explore them further over the next few days. So, let's take a high-level look at these concepts and get ready to take a deep dive soon.

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

Main Concepts

- Classes
- Objects
- Constructors
- Destructors
- Fields
- Properties
- Methods
- Inheritance
- Polymorphism
- Encapsulation
- 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

Classes

Institute of Technology Development of Canada

Classes

```
public class Car
{
    // Fields
    private int year;
    private int _speed;
    // Properties
    4 references
    public string Make{get; set;}
    4 references
    public string Model { get; set; }
    0 references
    public int Year
    {
        get { return year; }
        set { year = value; }
    }
    // Constructor
    0 references
    public Car(string make, string model, int year)
    {
        this.Make = make;
        this.Model = model;
        this.year = year;
    }
    // Methods
    0 references
    public void Start()
    {
        MessageBox.Show($"Starting the {Make} {Model}...");
    }
    0 references
    public void Stop()
    {
        MessageBox.Show($"Starting the {Make} {Model}...");
    }
    0 references
    public void Accelerate(int increment)
    {
        _speed += increment;
        MessageBox.Show($"Accelerating the {Make} {Model}... Current speed: {_speed} mph");
    }
}
```

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

- In Object-Oriented Programming (OOP), classes act like a blueprint that defines the properties and methods of an object. Properties are like variables that hold data related to the object, while methods are like functions that perform actions related to the object. By using classes, programmers can organize code into manageable components, making it easier to understand and manipulate.

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

- Think of a class like the blueprint for a car. Just as a blueprint outlines the design and specifications for a car, a class outlines the properties and methods for an object. The properties are like the parts that make up the car, such as the engine, wheels, and seats. Methods are like the actions that the car can perform, such as accelerating, braking, or turning. By using classes in OOP, programmers can build and modify cars (objects) more easily and efficiently, without having to reinvent the wheel each time.

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

Objects

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

Objects

- Next, we have objects. An object is an instance of a class, meaning it's a specific occurrence of the characteristics and behaviors defined in the class. Going back to our car class, an object of the car class might be a specific car such as a red 2024 Honda Civic.

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

Objects

```
0 references
public void myDemo()
{
    Car myCar = new Car("Toyota", "Camry", 2022);
    myCar.Start();
    myCar.Accelerate(10);
}
```

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

Properties And Fields

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

Fields vs Properties

- Both properties and fields can be used to store data in an object, but they have some important differences.

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

Fields vs Properties

```
// Properties
4 references
public string Make { get; set; }
4 references
public string Model { get; set; }
0 references
public int Year
{
    get { return year; }
    set { year = value; }
}
```

```
// Fields
private int year;
private int _speed;
```

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

Fields vs Properties

- Fields are variables that are directly defined within a class. They can be accessed directly by any method within the class but cannot enforce any constraints or perform any actions when their value is accessed or changed. They are typically used to store private implementation details that should not be exposed to the rest of the code.

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

Fields vs Properties

- Properties, on the other hand, are special methods that provide controlled access to a class's fields. They allow us to enforce constraints and perform actions when the data is accessed or modified. For example, we can use properties to ensure that a value falls within a certain range, or to update related fields when a value is changed.
- Properties in C# are like the doors and windows of a house. They provide a way to interact with the inside and outside of the house, while also controlling the flow of air, light, and sound. Just like how we can lock or unlock a door to control access to a room, we can set rules for the data contained within a property to control who can read or write to it.

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

Methods

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

Methods

```
1 reference
public void Start()
{
    MessageBox.Show($"Starting the {Make} {Model}...");
}
0 references
public void Stop()
{
    MessageBox.Show($"Starting the {Make} {Model}...");
}
1 reference
public void Accelerate(int increment)
{
    _speed += increment;
    MessageBox.Show($"Accelerating the {Make} {Model}... Current speed: {_speed}
      mph");
}
```

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

Methods

- A method is a block of code that performs a specific task. It's kind of like a recipe - you give it some ingredients (input) and it gives you a result (output).
- In C#, methods are defined within a class, just like properties. They can take input parameters and return output values, or they can simply perform some action without returning anything.
- Let's look at an example. Say we have a Car class with properties for make, model, and year. We could create methods called Start, Stop, and Accelerate that would display a message box.

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

Constructor

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

Constructors

- In C#, a constructor is a special method that is called when an instance of a class is created using the “new” keyword. It's used to initialize the object's properties or fields with values, allowing you to set up the object's initial state.
- Think of a constructor like a factory that produces a new car. When the car is first built, it needs to have certain things initialized, like its make, model, color, and year. These are the car's properties. The constructor is like the factory worker who puts these initial values into the car's properties, so that it's ready to be driven off the lot.

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

Constructors - Default

- A default constructor is a constructor that is provided by the compiler when a class does not have any constructor defined. A default constructor does not take any parameters and initializes all the fields to their default values. For example, in the following code, the Car class has a default constructor that initializes the fields to their default values:

```
// Default Constructor
0 references
public Car()
{
    this.Make = "";
    this.Model = "";
    this.year = 0;
}
```

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

Constructors - Overloaded

- An overloaded constructor is a constructor that takes one or more parameters. An overloaded constructor is used to provide different ways to create an object of a class with different initial values. For example, in the following code, the Car class has an overloaded constructor that takes three parameters to initialize the fields:

```
// Overloaded Constructor
1 reference
public Car(string make, string model, int year)
{
    this.Make = make;
    this.Model = model;
    this.year = year;
}
```

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

Constructors - Copy

- A copy constructor is a constructor that takes an object of the same class as a parameter and creates a new object with the same values. A copy constructor is used to create a new object with the same values as an existing object. For example, in the following code, the Car class has a copy constructor that takes a Car object as a parameter and creates a new Car object with the same values:

```
// Copy Constructor
0 references
public Car(Car car)
{
    this.Make = car.Make;
    this.Model = car.Model;
    this.year = car.year;
}
```

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

Destructors

- Destructors are not required in C# because the .NET runtime automatically manages memory for objects, and cleans up objects when they are no longer in use. However, they can be useful for performing custom cleanup actions or releasing resources, and should be used when necessary.