**Software Developer Job Ready Programme**

**Agenda Week 1**

**Day1**

Object-Oriented prog ramming Four basic principles

**Day2**

What is a Class

Value types versus Reference types Class declaration

Constructors and initialization

**Day3**

SOLID Principles

Why SOLID Principles

**Object-Oriented program m ing (C#)**

Anyfoolcanwritecodethatacomputercan understand.Goodprogrammerswritecode thathumanscanunderstand.

- Martin Fowler

Refactoring: Improving the Design of Existing Code by Martin Fowler, Kent Beck (Contributor), John Brant (Contributor), William Opdyke, don Roberts

**C# is an object-oriented program m ing language. The four basic principles of object-oriented**

**Abstraction** **Encapsulation** **Inheritance** **Polym orphism**

**Abstraction**

In C#, abstraction is implemented using abstract classes and methods.

An abstract class is a class that cannot be instantiated, and must be inherited by a derived class in order to be used.

**Encapsulation**

Encapsulation is the hiding the internal state and functionality of an object and only allowing access through a public set of functions.

It refers to the bundling of data and related operations into a single unit, or object. In C#, encapsulation is achieved through the use of classes and access modifiers such as public, private, and protected.

**Inheritance**

Ability to create new abstractions based on existing abstractions.

Conceptually, a derived class is a specialization of the base class. For example, if you have a base class Animal, you might have one derived class that is named Mammal and another derived class that is named Reptile. A Mammal is an Animal, and a Reptile is an Animal, but each derived class represents different specializations of the base class.

**Polymorphism**

Ability to implement inherited properties or methods in different ways across multiple abstractions.

Polymorphism is a Greek word that means "many-shaped"

That is, the same entity (method or operator or object) can perform different operations in different scenarios.

**Polymorphism**

Ability to implement inherited properties or methods in different ways across multiple abstractions.

Polymorphism is a Greek word that means "many-shaped"

That is, the same entity (method or operator or object) can perform different operations in different scenarios.

**Advantages of OOP**

**Simplifying debugging**

**Harnessing code reusability for enhanced efficiency Amplifying productivity through OOP practices Streamlining data management to reduce redundancy Problem-solving capabilities**

**Enhanced security measures**

**Improved code structuring and organisation**

**Introduction to C#**

**The C# language is the most popular language for the** [**.NET platform,**](https://learn.microsoft.com/en-us/dotnet/csharp/) **a free, cross-platform, open source development environment. C# programs can run on many different devices, from Internet of Things (IoT) devices to the cloud and everywhere in between. You can write apps for phone, desktop, and laptop computers and servers.**

**C# programs consist of one or more files. Each file contains zero or more namespaces. A namespace contains types such as**

**classes structs interfaces enumerations delegates**

**Helpful links**

**A tour of the C# language - https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/overview**

**Fundamentals for beginners - https://learn.microsoft.com/en-us/shows/csharp-fundamentals-for-absolute-beginners/**