

VERSION

2018

CRASH COURSE

SR. CODE

EAPL/CRASH/CRTC13

COURSE CODE

EACCS

SUB CATEGORY

SOFTWARE DEVELOPMENT


TOTAL DURATION
45
HOURS


THEORY TAKEN
12
HOURS


PRACTICAL TAKEN
33
HOURS

ELYSIUM
ACADEMY

CORE C#
PROGRAMMING

**ELYSIUM
ACADEMY**

**CORE C#
PROGRAMMING
COURSE**

ELYSIUM
ACADEMY

CORE C#
PROGRAMMING
COURSE

COURSE DESCRIPTION



The Core C# Programming course immerses learners in the fundamentals of C# language, covering topics such as syntax, data types, and object-oriented programming principles. Practical exercises reinforce theoretical knowledge, empowering students to build functional applications.

COURSE GOALS



Upon completion, you will confidently navigate C# development, mastering concepts like LINQ, asynchronous programming, and MVC architecture. They will acquire the skills to develop scalable, maintainable software solutions for diverse industry needs.

FUTURE SCOPE



Graduates will be well-prepared for roles as C# developers, software engineers, or application architects. They will possess the versatility to adapt to various domains, including web development, desktop applications, game development, and more, ensuring long-term career prospects in the dynamic software industry.

01

CHAPTER

INTRODUCTION

01. INTRODUCTION TO C#

- a. Strong Programming Features of C#
- b. The .Net Framework
- c. Integrated Development Environment (IDE) for C#
- d. Writing C# Programs on Linux or Mac OS
- e. Creating Hello World Program
- d. Compiling and Executing the Program
- f. C# Keywords

02. BASIC SYNTAX

- a. The using Keyword
- b. The class Keyword
- c. Comments in C#
- d. Member Variables
- e. Member Functions
- f. Instantiating a Class
- g. Identifiers
- h. C# Keywords

03. DATA TYPES

- a. Value Type
- b. Reference Type
- c. Object Type
- d. Dynamic Type
- e. String Type
- f. Pointer Type



2.5
HRS



08
HRS

O4. TYPE CONVERSION

C# Type Conversion Methods

O5. VARIABLES

- a. Defining Variables
- b. Initializing Variables
- c. Accepting Values from User
- d. Lvalue and Rvalue Expressions in C#

O6. CONSTANTS AND LITERALS

- a. Integer Literals
- b. Floating-point Literals
- c. Character Constants
- d. String Literals
- e. Defining Constants

O7. OPERATORS

- a. Arithmetic Operators
- b. Relational Operators
- c. Logical Operators
- d. Bitwise Operators
- e. Assignment Operators
- f. Miscellaneous Operators
- g. Operator Precedence in C#

02

CHAPTER

DECISION MAKING & CONTROLLING

01. DECISION MAKING

- a. if Statement
- b. if...else Statement
- c. The if...else if...else Statement
- d. Nested if Statements
- e. Switch Statement
- f. The ? : Operator

02. LOOPS

- a. While Loop
- b. For Loop
- c. Do...While Loop
- d. Nested Loops
- e. Loop Control Statements
- f. Infinite Loop


1.5
HRS


3.5
HRS

03

CHAPTER

FUNCTIONS & METHODS

01. METHODS

- a. Defining Methods in C#
- b. Calling Methods in C#
- c. Recursive Method Call
- d. Passing Parameters to a Method
- e. Passing Parameters by Value
- f. Passing Parameters by Reference
- g. Passing Parameters by Output


2.5
HRS


1.5
HRS

04

CHAPTER

ARRAYS

01. ARRAYS

- a. Declaring Arrays
- b. Initializing an Array
- c. Assigning Values to an Array
- d. Accessing Array Elements
- e. Using the foreach Loop
- f. C# Arrays
- g. Multidimensional Arrays
- h. Two-Dimensional Arrays
- i. Jagged Arrays
- j. Passing Arrays as Function Arguments
- k. Param Arrays
- l. Array Class
- m. Properties of the Array Class
- n. Methods of the Array Class


1.5
HRS


3.5
HRS

05

CHAPTER

OOPS

01. CLASSES

- a. Defining a Class
- b. Member Functions and Encapsulation
- c. C# Constructors
- d. C# Destructors
- e. Static Members of a C# Class


2.5
HRS


05
HRS

O2. ENCAPSULATION

- a. Public Access Specifier
- b. Private Access Specifier
- c. Protected Access Specifier
- d. Internal Access Specifier

O3. INHERITANCE

- a. Base and Derived Classes
- b. Initializing Base Class
- c. Multiple Inheritance in C#

O4. POLYMORPHISM

- a. Static Polymorphism
- b. Dynamic Polymorphism

O5. OPERATOR OVERLOADING

- a. Implementing the Operator Overloading
- b. Overloadable and Non-Overloadable Operators

O6. INTERFACES

- a. Declaring Interfaces

O7. NAMESPACES

- a. Defining a Namespace
- b. The using Keyword
- c. Nested Namespaces

06

CHAPTER

EXCEPTION HANDLING

01. EXCEPTION HANDLING

- a. Exception Classes in C#
- b. Handling Exceptions
- c. Creating User-Defined Exceptions
- d. Throwing Objects

02. FILE I/O

- a. C# I/O Classes
- b. The FileStream Class Advanced File Operations in C#
- c. Reading from and Writing to Text Files
- d. The StreamReader Class
- e. The StreamWriter Class
- f. Reading from and Writing into Binary files
- g. The BinaryWriter Class
- h. Windows File System
- i. The DirectoryInfo Class
- j. The FileInfo Class


02
HRS


3.5
HRS

07

CHAPTER

REFLECTION & PROPERTIES

01. REFLECTION

- a. Applications of Reflection
- b. Viewing Metadata

02. PROPERTIES

- a. Accessors
- b. Abstract Properties


30
MINS


1.5
HRS

08

CHAPTER

INDEXERS & DELEGATES

O1. INDEXERS

- a. Use of Indexers
- b. Overloaded Indexers

O2. DELEGATES

- a. Declaring Delegates
- b. Instantiating Delegates
- c. Multicasting of a Delegate
- d. Using Delegates



01
HR



01
HR

09

CHAPTER

EVENTS & COLLECTIONS

O1. EVENTS

- a. Using Delegates with Events
- b. Declaring Events

O2. COLLECTIONS

- a. ArrayList Class
- b. Hashtable Class
- c. SortedList Class
- d. Stack Class
- e. Queue Class
- f. BitArray Class



01
HR



2.5
HRS

Placement Assistance

100%

135+ Professional Courses

Practical Sessions

90%

67+ Global Pacts

Corporate Placements

65%

170+ IT Companies Tie-Up

ELYSIUM
GROUP OF
COMPANIES

**ELYSIUM
ACADEMY**

**PRIVATE
LIMITED**

AUTHORIZED INTERNATIONAL

Partners

