C#

Execution process

Native code (eg. c# code)

MSIL (Microsoft intermediate language) or CIL (common intermediate language)

Machine code (binary code)

Data Types,variables,constants & readonly

Variables: Variable is a holder that holds data.

Data Types: Defines which kind of data you are going to store in a variable.Some common data types are:
Int,float,double,decimal,char,bool,datetime

Constants: a data that can not be changed

Readonly: is constant but can be changed from constructor.

Type conversion

Implicit conversion: Implicit conversion is possible only when the value is kind of similar to the new type and not going to truncate or round off after storing it into another variable.
int a= 445678 => Long l= a;
Int a = 6; => Double b=a;
Double c=3.222;=> Int d=c; // not possible

Explicit conversion: If a conversion can not be made without risk of losing data then we can convert it explicitly.
double d=1234.22; int a = (int) d;

Operators

Operator is a symbol that tells compiler to perform a mathematical or logical operation.

Arithmetic: +, - , *, /, %,++,--
Relational: ==, !=, >,<, >=, <=
Logical: &&, ||, !
Bitwise: perform bitwise operations
Bitwise complement (~), bitwise and (&), bitwise or (|), bitwise xor(^)
Assignment: =,+=,-=,*=,/=,

Conditional statements

Conditional statements are used to execute your code in some certain conditions.

- If
- if..else
- if..Else if..
- switch

Loops

It is used to perform iterations.

- While
- Do while
- For
- foreach

Break ,continue and goto

Break jumps you out of the loop.
Continue statement breaks one iteration.
Goto: Jumps to certain part of program

String

String is an object that stores text value.
String str = “hello”;

Part9: Exercise 1

1.Enter the 3 inputs and multiply them. Eg: a=5,b=4,c=6, d=5*4*6
2. Enter the 2 numbers, print their average
3. If someone enters negative value then show the error message “We do not accept negative value”.
4. Print two stars ** 4 times
5. Find the number of characters in “JOHN DOE”
6. Print numbers from 1 to 10 except 6 (o/p 1 2 3 4 5 7 8 9 10)

Part10: Exercise 2

1. Print this pattern

   *

   **

   ***

   ****

   *****

1. Check the number is even or odd

Part11: Exercise 3 (snake,water,gun game)

Two player game
Inputs : ‘s’ / ‘w’ / ‘g’
Rules:
Snake vs water: snake drinks water so snake wins
Sake vs gun : gun kills snake so gun wins
Water vs gun: gun drowns in water so water wins

Part 12: Object oriented programming (class,object,namespace)

Class : A class is a blueprint that has certain attributes and features.
Suppose a car is a class. Make,model,color,year and price.It can have functionalities like start,stop, run and move.

Object: When we create an instance of the class,it is called object

Namespace: It is a scope that contains the related set of objects.
Example: School namespace can contain several classes like course,student,department,marks etc.
Within a namespace you can contain the following types:
class,Interface,struct,enum,delegate and a nested namespace (namespace inside a namespace)
Example: System is a inbuilt namespace:

Part 13: Constructor

It is a special method that is invoked when an object of the class gets instantiated.
Means when an object gets memory,this method is invoked.
=>It has the same name as Class
=>Does not contain any return type
=> Constructor has same name as Class Name and doent't contain any return type.
=> ***ctor*** is short form to create a constructor for a class.

Default constructor(): It is the first method that is going to invoked in a class.
Parameterized constructor(): You can also provide parameters to the constructor.

Part 14: Exercises for classes and methods

1. Create a class employee with name,salary,dateOfJoining attributes,a method getEmployeeDetails() . Name,salary and date of joining will bet set from constructor.
2. Create a class that contains Add(),sub(), mul(),div() methods,those method should return a desired value.
3. Create a method that checks number is even or odd.

Part 15 : Inheritance

It is a mechanism by which one class can access the functionality of another class. The class that shares its functionality is called base class and accessor class is called derived class.

Part 16: Method overloading and method overriding

Polymorphism : Multiple methods with same name:q

1.Static polymorphism : Called method is determined at compile time
(Method overloading and operator overloading)

2.Dynamic polymorphism:Called method is determined at run time
(Method overriding)

Method overloading: Creating multiple methods with same name but different signature.

Method overriding: Multiple methods with same name same signatures

Part 17: Access modifier

Defines the accessibility scope of the member

- **Public:** Can be accessed anywhere.
- **Private**: Access is limited to the type.
- **Protected**: Can be accessed only to the derived classes.
- **Internal:** Accessible to the current assembly.
- **Protected internal:** Can be accessed from all the classes in the same assembly.But if you are trying to access it from another assembly then you need to inherit that class.
- **Private protected:** It is similar to protected, the class itself and its derived class can access its members..

Part 18: Properties

Properties are the members of a class that is used to access the data. It is useful to provide Validation logic to the class.

Public string Name {get;set;}

Part 19: Static class and static members

Static class: Static class is similar to normal class but you can not create its instance.

- It is sealed
- Can not instantiated
- Can contain only static members.

Static members: Static members can be declared inside non static class.

Static variable: Single copy of variable is created at shared between all the objects.

Part 20: Abstract classes

Abstract class is a incomplete class,it means you can not create a instance of it,it can be used as base class only.

- Can’t instantiated
- Can contain abstract and not non abstract method

Part 21: Interface

Interfaces are blueprints for class.
=>interfaces are very useful when we are writing test cases.

Part 22: Exception handling

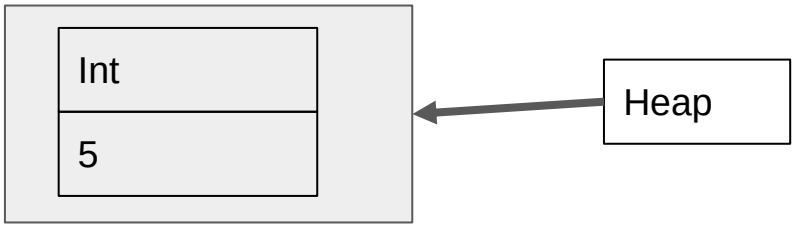
try,catch,finally

Part 23: Array

Array is a collection of similar type of data.
Eg. int[] arr, char[] cArr

Part 24: Boxing and unboxing

Boxing : converting value type to object type .
Int a=5;
Object obj = a;



Object a

Unboxing: converting object type to value type

Value types: resides in the stack .eg int,bool,char etc.
Object type: object

Part 25: Collections

Non-generic collection (System.Collections classes)

- ArrayList
- Hashtable : (key,value) =>key should be unique
- Queue
- Stack

Generic collections (System.Collections.Generic classes)
(strongly typed)=>means they are type safe, no need of boxing unboxing, faster.

- Dictionary<TKey,TValue>
- List<T>=> List<int>
- Queue<T>
- Hashset<T> : contains unique items only , can't be sorted
- SortedList<TKey,TValue>

Part 26: Extension methods

Allows us to add functionality to existing type

Part 27: Enums

Enum is the collection of constants

Part 28: Indexer

Allows objects to behave like array.

Part 29: Delegates

- **Delegates** are the types, which holds the reference of a method or multiple methods.
- **Delegates** are useful to achieve **callback mechanism** in c#.

Anonymous methods:

Part 30: Generic Delegates (Action,Func & Predicate)

Action<T>: Takes 0 or more inputs *does not return* anything.
Func<T1,T2> : Takes 0 or more inputs but *always return* a value.
Predicate<T>: Takes 1 input and returns a boolean value.

Part 31: Multithreading

It is a techniques that allows to perform multiple operations simultaneously.
Your tasks are divided between threads so that they can perform simultaneously.

Part 32: Asynchronous tasks

Async task is a non blocking task. If you have a time consuming task,you can create it as a async task. So that you can move further and don’t wait to finish the task.That task will run in the background.
Async task can be multithreaded or single threaded.