# Modulo 1

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Nome | Francesco |
|  |  | Cognome | Dell’Ascenza |
|  |  | Data | 26/03/2021 |

Leggete attentamente ogni domanda e argomentare quanto più possibile **fornendo anche degli esempi**.  
ATTENZIONE: Le domande a risposta multipla possono contenere più risposte corrette.

1. *Dare una panoramica degli elementi principali nel Framework .Net. Cosa è successo con .Net 5?*
2. *Definire i principi su cui si basa il linguaggio C#*
3. *Dare una spiegazione di Garbage Collector.*
4. *Descrivere le differenze tra Reference Type e Value Type*
5. *Cosa è un delegate?*
6. *Scrivere una query LINQ : Date le gare disponibili, ricavare il tempo medio, massimo e minimo per disciplina.*
7. *Definire cosa è il design Pattern Factory*

*Mettere la prova pratica e teorica su Github.*

*1 Il framework .NET è un framework di sviluppo originariamente pensato per essere diviso riguardo lo sviluppo di application forms, web developement ad esempio con asp.net e mobile app, ma questo fino a .NET 5 che invece fa da unificatore che si prepone quindi di riunire in un unico framework*

*2 Il linguaggio C# è un linguaggio Object Oriented, quindi si basa sui principi della programmazione ad oggetti che sono la creazione di classi che rappresentano il template di oggetti che poi verranno istanziati, hanno proprietà e metodi. Altro principio è quello dell’astrazione ovvero una classe può essere astratta poiché non implementa tutti i metodi che vengono dichiarati ma saranno implementati nelle classi che ereditano da questa.*

*3 Il garbage collector è preposto all’eliminazione e ottimizzazione automatica dei dati in memoria*

*4 I value type sono i primitivi come int double bool e sono storati nello stack, ogni copia ad essi comporta una copia del dato contenuto in essi, al contrario i reference type sonoun riferimento storato nell’heap, ogni copia è una copia solo della reference, non del dato, sono ad esempio oggetti ma anche le stringhe.*

*5 Un delegate è un puntatore a funzione, un delegate infatti quando viene definito ritorna un type e prende dei parametri, quando viene richiamato poi gli si passa una funzione che ha le stesse caratteristiche ovvero ritorna lo stesso type e prende gli stessi parametri.*

*6 from result in results*

*Join competition in competitions*

*On result.ID == competition.ID*

*Group result.time by competition.discipline into disc*

*Select new {discipline = disc.Key, AvgTime = disc.average(t=>t.time), MaxTime = disc.Max(t=>t.time), MinTime = disc.Min(t=>t.time)}*

*7 Un design pattern è un’architettura comune, la Factory che è un design pattern di creazione è un’archittura in cui si ha un’interfaccia che viene implementata da diverse classi concrete, queste poi si interfacciano con il client tramite la factory che gestiste la richiesta di creazione di un oggetto di questa o l’altra classe in base ad una business logic. Ad esempio un utente può interfacciarsi con un software che in base a dei parametri che fornisce l’utente gli assegna una macchina, l’utente vede solo la macchina che gli viene assegnata inconsapevole di tutte le possibili macchine e delle loro proprietà. Questa gestione viene fatta tramite una factory.*