* **Cos’è L’OOP E QUALI SONO I SUOI PRINCIPI FONDAMENTALI?**

La OOP è un tipo di programmazione che consiste nella creazioni di oggetti che interagiscono tra loro, proprio come nella realtà.

I principi fondamentali della programmazione ad oggetti sono l’ereditarieà, il polimorfismo, l’incapsulamento e l’astrazione.

* **QUALI SONO LE DIFFERENZE TRA L’OOP E LA PROGRAMMAZIONE PROCEDURALE?**

La programmazione procedurale consiste nello scrivere le istruzioni in maniera sequenziale, mentre nella programmazione ad oggetti il codice è organizzato in oggetti che interagiscono tra loro.

* **COS’è UNA CLASSE IN OOP E QUALI SONO I SUOI COMPONENTI?**

Una classe è un modello di oggetto. I suoi componenti sono gli attributi, che descrivono le caratteristiche che avranno le sue istanze (gli oggetti) e i metodi, ovvero le funzionalità e le azioni che può eseguire.

* **QUAL È LA DIFFERENZA TRA UNA CLASSE E UN OGGETTO?**

La classe è il modello principale, mentre l’oggetto è una sua istanza, che avrà dei valori agli attributi.

* **QUAL E’ IL CONCETTO DI INCAPSULAMENTO IN OOP?**

Il concetto di incapsulamento consiste nel raggruppare attributi e metodi all’interno di una classe. Serve anche a proteggere i dati dall’accesso esterno per garantire che le operazioni sui dati avvengano in maniera controllata.

* **COS’è L’EREDITARIETA’ IN OOP E COME FUNZIONA?**

L’ereditarietà è quel concetto secondo cui una classe (classe derivata o sottoclasse) può ereditare attributi e metodi da un’altra classe detta superclasse (o classe base).

* **QUAL È IL CONCETTO DI POLIMORFISMO IN OOP?**
* **COS’è UNA RELAZIONE DI COMPOSIZIONE IN OOP?**

Per relazione di composizione si intende una relazione per cui una classe non può esistere senza l’altra (appunto una classe è composta dall’altra classe)

* **QUAL È LA DIFFERENZA TRA UNA RELAZIONE DI COMPOSIZIONE E UNA RELAZIONE DI AGGREGAZIONE IN OOP?**
* **QUAL È IL CONCETTO DI INTERFACCIA IN OOP E PERCHè è IMPORTANTE?**
* **QUALI SONO I VANTAGGI DELL’OOP RISPETTO AD ALTRI PARADIGMI DI PROGRAMMAZIONE?**
* **QUALI SONO LE DIFFERENZE TRA UN LINGUAGGIO DI PROGRAMMAZIONE ORIENTATO AGLI OGGETTI E UNO NON ORIENTATO AGLI OGGETTI?**
* **QUAL È LA DIFFERENZA TRA UN’INTERFACCIA E UNA CLASSE ASTRATTA?**
* **UNA CLASSE ASTRATTA E’ IMPLEMENTABILE DA UNA SOTTOCLASSE?**

No, una classe astratta non è implementabile da una sua sottoclasse.

* **COME SI UTILIZZA IL CONCETTO DI POLIMORFISMO IN OOP PER SCRIVERE CODICE PIU’ MODULARE E FLESSIBILE?**
* **COME SI PUO IMPLEMENTARE UNA RELAZIONE DI EREDITARIEA’ IN OOP?**
* **COS’è UN COSTRUTTORE IN OOP E QUANDO VIENE CHIAMATO?**
* **QUALI SONO I TIPI DI ACCESSO IN OOP E QUAL E’ LA LORO IMPORTANZA?**
* **COS è IL POLIMORFISMO IN OOP?**
* **QUAL È LA DIFFERENZA TRA UN’ISTANZA DI UNA CLASSE E UNA VARIABILE LOCALE?**
* **COS’è UN’ECCEZIONE IN OOP E COME GESTIRLA?**
* **COME SI UTILIZZA L’EREDITARIETA’ MULTIPLA IN OOP E QUALI SONO LE SUE POTENZIALI PROBLEMATICHE?**
* **COME SI UTILIZZA LA PAROLA CHIAVER SUPER IN OOP E IN QUALI SITUAZIONI POTREBBE ESSERE NECESSARIA?**
* **COME SI UTILIZZA LA PAROLA CHIAVE THIS IN OOP E IN QUALI SITUAZIONE POTREBBE ESSERE NECESSARIA?**
* **QUAL È IL CONCETTO DI BINDING IN OOP E QUALI SONO I SUOI TIPI?**
* **QUAL È IL CONCETTO DI GARBAGE COLLECTOR IN OOP E COME FUNZIONA?**

Il garbage collector si occupa di gestire la memoria pulendola ed eliminando quelle variabile che non vengono più utilizzate nel nostro programma.