# Test di valutazione – Modulo 1

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Nome | Tiziana |
|  |  | Cognome | Camboni |
|  |  | Data | 05/02/2021 |

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. *Spiegare le differenze tra memoria principale, secondaria e virtuale*

*Per memoria virtuale si intente la capacità di un sistema operativo di far sembrare la memoria maggiore si quella che è fisicamente.*

*La memoria principale è una memoria volatile come la RAM , mentre la memoria secondaria è data da un salvataggio permanente. In particolare è sulla disponibiltà della prima che si sfrutta la memoria virtuale.*

1. *Cosa significa che un linguaggio è tipizzato? Spiegare la suddivisione in tipi in C# e le relative caratteristiche.*

*Per linguaggio tipizzato si intende il fatto che le variabili utilizzate hanno una definizione, un “tipo” che le caratterizza e ne attribuisce delle proprietà.*

* *Int -> definisce numeri interi;*
* *Float -> definisce numeri in virgola mobile;*
* *String -> definisce una stringa di caratteri;*
* *Array -> definisce un serie di numero fisso di elementi;*
* *ArrayList -> definisce una serie di numeo variabili di elementi;*
* *Classe -> viene definito un tipo specifico con campi, proprietà e metodi.*

1. *Cos’è un sistema operativo? Dare una spiegazione delle funzionalità.*

*Un Sistema operativo è un insieme si software che permette la gestione delle funzionalità di un computer.*

* *Gestione Interfaccia utente (es. Terminale)*
* *Gestione memoria (sfruttando principio di località e memoria virtuale)*
* *Gestione I/O (periferiche)*
* *Gestione Processore (attraverso il kernel che garantisce la sicurezza di non sovrascrivre parti di memoria dedicate a determinati processi)*

1. *Descrivere il funzionamento degli operatori logici e delle relative tabelle di verità.*

*Gli operatori logici sono degli strumenti che permettolo le operazioni tra bit e le relative tabelle di verità vengono in aiuto per ricordare il risultato delle operazioni.*

1. *Spiegare la differenza tra ricorsione e iterazione e le relative peculiarità.*

*La ricorsione consiste nel richiamare all’interno di una funzione, la funzione stessa; l’interazione consiste nel svolgere un codice di blocco finchè una codizione è valida (cicli while e for);*

*Nella pratica, la prima risulta piu affidabile nel caso in cui i richiami alla stessa funzione vengono svolti un numero molto limitato di volte (es. N<10), altrimenti risulta più affidabile la seconda.*

1. *Dare una definizione di casting. Fornire degli esempi.*

*Il casting è un tipo di conversione di un tipo in un altro affine e può essere fatta con le parentesi tonde. Può essere fatta in modo:*

* *Implicito: int c= 3;*

*decimal d= (int) c;*

* *Esplicito: int c= 3;*

*d= convert.ToDecimal(c);*

1. *Spiegare cosa è una routine. Specificare le tipologie di routine e fornire degli esempi per ciascuna.*

*La routine è un insieme di procedure e funzioni che costituiscono un codice di blocco che vengono richiamate in altre parti del codice.*

1. *Descrivere la funzione del compilatore*

*Il compilatore, controllando la sintassi di un codice sorgente ne permette il passaggio a codice macchina.*

1. *Descrivere le funzioni di un sistema di versionamento e le diverse tipologie.*

*La funzione principale di un sitema di versionamento è mantere la memoria delle modifiche di un codice.*

* *Versione Locale: salvataggio nella memoria dell’utente (non condivisbile)*
* *Versione Centralizzati: salvataggio in remoto in un server (condivisibile ma rischioda in caso di down della rete)*
* *Versioni Distribuite: il salvataggio viene fatto sia in remoto che nel server (condivisibile ma possono esserci problemi di incompatibilità nel momento in cui due utenti caricano una versione aggiornata)*

*Esercizio Pratico*

Creare una Console Application che gestisca i Task dell’utente.

Per Task viene inteso un oggetto che ha una descrizione, una data di scadenza e un livello di importanza (Basso, Medio, Alto).

L’utente può:

* Vedere i Task inseriti
* Aggiungere un nuovo Task
* Eliminare un Task
* Filtrare i Task per importanza

Requisiti Tecnici:

-Salvare i Task in un file

-Utilizzare adeguatamente il concetto di classe

-Dividere le funzionalità in relative funzioni e procedure

-Commentare

-Mettere una nomenclatura conforme

-Le date di scadenza devono essere posteriori o uguali rispetto alla data di inserimento

-Controllare l’input utente

Opzionale: Utilizzare Enum

Mettere il codice dell’esercizio in un Repository di GitHub.