

Date: _____

BCSF20A004

MTWTF

Enterprise Advance Development

Assignment # 02

(i)

Generic Collections :-

Generic collections are very strong in types, the type must be declared at the compile time i.e. string, double, int, etc.

Generic collections are better in performance, executing and efficiency. Boxing and unboxing are not involved in this. It promote code reusability as the same collection type can be used with different data types, enhancing code maintainability and reducing redundancy. They are flexible and can be easily extended to handle various data types.

list, Dictionary, Queue, Stack and HashSet are generic collections.

Non-Generic Collections :-

Non-generic collections are not strongly typed and can hold objects of any type. It suffers from performance suspended due to boxing and unboxing value while executing different value types, which result in low performance. It may need duplicate code or manual typecasting to work with various data types, which leads to less code reusability. It may lack the flexibility to adapt to diff data types.

Arraylist, Hashtable and Bitarray are non-generic collections.

(ii)

According to non-generic collections specifically is better to use during our needs. As we can make an ArrayList obj and save different information of different datatypes in one obj. Example we have to save name (dt = String), a number, address, email, cprn (dt = float) of a student so in this we need to collection which can save all of info of different data types. So, it will be easily manage info of person if its saved in single obj of non-generic. On other, if we have fixed data types entries, then we have to save data of single to into different components.

So, it's not suitable to use
fixed datatypes entities form
design.