|  |
| --- |
| **Day-6 morning assignment**  **By**  **Bhanu Rama Krishna Prakash Jakkamsetti**  **31/1/2022** |

|  |
| --- |
| 1. Create a simple program to declare Array List and assign some values  and find sum. |
| Code: |
| using System;  using System.Collections;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace ArrayList\_and\_assign\_some\_values  {  internal class Program  {  static void Main(string[] args)  {  ArrayList data = new ArrayList();  int sum = 0;  data.Add(5);  data.Add(10);  data.Add(20);  data.Add(30);  data.Add(35);  foreach (var d in data)  sum = sum + (int)d;  Console.WriteLine($"sum of ArrayList is {sum}");  Console.ReadLine();  }  }  } |
| Output: |
|  |

|  |
| --- |
|  |

2.Research and find how the values of Array List are stored in the memory.

* The elements of an ArrayList are stored in a chunk of contiguous memory. When that memory becomes full, a larger chunk of contiguous memory has to be allocated (usually twice the size) and the existing elements are copied into this new chunk. We call this chunk the capacity of the ArrayList object.

3.What are the dis-advantages of Array List (Collections Array List)

* If there is a chance of assigning wrong value we get runtime error.
* Every time we unbox and do operation.

|  |
| --- |
| 4. Create a simple program to declare List<int> and assign some values and find sum. |
| Code: |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace declare\_List\_and\_assign\_some\_values  {  internal class Program  {  static void Main(string[] args)  {  List<int> data = new List<int>();  int sum=0;  data.Add(1);  data.Add(2);  data.Add(3);  data.Add(4);  data.Add(5);  data.Add(6);  foreach (int i in data)  sum=sum+i;  Console.WriteLine($"List of sum is {sum}");  Console.ReadLine();  }  }  } |
| Output: |
|  |

5.In a tabular format write the differences between Collections and generics.

|  |  |  |
| --- | --- | --- |
|  | Collection(Array List) | Generics(List) |
| Name space | System.collections; | System.Collections.Generic; |
| Each element is of what type | Object type | Any data type  List<T> |
| Type casting | We need type casting | We doesn’t need type casting |
| Example | ArrayList data=new ArrayList(); | List<int> data=new List<int>(); |

6.Research and find how the values of List<T> are stored in the memory.

* In a List<T> , the memory to store the value types is within the memory allocated for the System. Array (i.e. "Over Here"). In an ArrayList each element is just a reference to a boxed value type, so the actual memory to store each value type is elsewhere on "The Heap", i.e. somewhere "Over There".

|  |
| --- |
| 7. WACP to declare List<String> and add 5 values and print the values using  a. for loop  b. foreach loop  c. Lambda Expression |
| Code: |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace ListString\_and\_add\_5\_values  {  internal class Program  {  static void Main(string[] args)  {  List<string> data = new List<string>();  data.Add("krish");  data.Add("bhanu");  data.Add("meg");  data.Add("dhivi");  //using forloop  for (int i = 0; i< data.Count; i++)  Console.WriteLine(data[i]);  //using foreach loop  foreach (var d in data)  Console.WriteLine(d);  //using lambda  data.ForEach(d => Console.WriteLine(d));  }  }  } |
| Output: |
|  |

|  |
| --- |
| 8. WACP to declare List<int> and read 5 values from user and find sum using  a. for loop  b. foreach loop  c. Lambda Expression |
| Code: |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Listint\_and\_read\_5\_values  {  internal class Program  {  static void Main(string[] args)  {  List<int> data = new List<int>();  int sum1 = 0,sum2 = 0,sum3 = 0;  data.Add(1);  data.Add(2);  data.Add(3);  data.Add(4);  data.Add(5);  data.Add(6);  //using for loop  for (int i = 0; i < data.Count; i++)  sum1=sum1+data[i];  Console.WriteLine($"by using forloop {sum1}");  //using foreach loop  foreach (var d in data)  sum2=sum2+d;  Console.WriteLine($"by using foreach loop {sum2}");  //using lambda  data.ForEach(d => sum3 = sum3 + d ) ;  Console.WriteLine($"by using lambda {sum3}");  Console.ReadLine();  }  }  } |
| Output: |
|  |

9.In a tabular format write all data types in C# and write the respective alias name

|  |  |  |
| --- | --- | --- |
| SNo | Datatype | Alease name |
| 1 | byte | Byte |
| 2 | ushort | UInt16 |
| 3 | uint | Uint32 |
| 4 | ulong | Uint64 |
| 5 | sbyte | SByte |
| 6 | short | Int16 |
| 7 | int | Int32 |
| 8 | long | Int64 |
| 9 | float | Single |
| 10 | double | Double |
| 11 | decimal | Decimal |
| 12 | bool | Boolean |
| 13 | char | Char |
| 14 | string | String |

|  |
| --- |
| 10. Write example programs for implicit and explicit type casting. |
| Code: |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace implicit\_and\_explicit\_type\_casting  {  internal class Program  {  static void Main(string[] args)  {  //implicit type casting  short a = 10;  int b = a;  Console.WriteLine(b);  //explicit type casting  int c = 20;  short d = (short)c;  Console.WriteLine(d);  }  }  } |
| Output: |
|  |