

.NET LAB PROGRAMS SOLUTION

Subject Code: 10MCA57

I.A Marks: 50

Hours/Week: 3

Exam Hours: 03

Total Hours: 42

Exam Marks: 50

1. Write a Program in C# to Check whether a number is Palindrome or not.

Palindrome.cs

```
1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Text;
5.
6. namespace prog1
7. {
8.     class Palindrome
9.     {
10.         static void Main(string[] args)
11.         {
12.             int num, temp, digit, rev = 0;
13.             Console.WriteLine("Enter a number :");
14.             num=int.Parse(Console.ReadLine());
15.             temp = num;
16.             while (temp > 0)
17.             {
18.                 digit = temp % 10;
19.                 rev = rev * 10 + digit;
20.                 temp = temp / 10;
21.             }
22.             if (num == rev)
23.                 Console.WriteLine(num + " is palindrome number.");
24.             else
25.                 Console.WriteLine(num + " is not palindrome number.");
26.             Console.ReadLine();
27.         }
28.     }
29. }
```

2. Write a Program in C# to demonstrate Command line arguments processing.

CmdLineArg.cs

```
1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Text;
5.
6. namespace prog2
7. {
8.     class CmdLineArg
9.     {
10.         static void Main(string[] args)
11.         {
```

.NET LAB PROGRAMS SOLUTION

```
12.         if (args.Length == 0)
13.             Console.WriteLine("There is no Command Line Argument defined");
14.
15.         else
16.             for (int i = 0; i < args.Length; i++)
17.                 Console.WriteLine(Convert.ToString(args[i]));
18.         Console.ReadLine();
19.     }
20. }
21. }
```

3. Write a Program in C# to find the roots of Quadratic Equation.

QuadraticEquation.cs

```
1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Text;
5.
6. namespace prog3
7. {
8.     class QuadraticEquation
9.     {
10.         static void Main(string[] args)
11.         {
12.             int a, b, c;
13.             double disc, denom, X1, X2;
14.             Console.WriteLine("Enter the value of a, b & c");
15.             a = int.Parse(Console.ReadLine());
16.             b = int.Parse(Console.ReadLine());
17.             c = int.Parse(Console.ReadLine());
18.             disc = (b * b) - (4 * a * c);
19.             denom = (2 * a);
20.             if (disc > 0)
21.             {
22.                 Console.WriteLine("The Roots are Real roots...");
23.                 X1 = (-b / denom) + (Math.Sqrt(disc) / denom);
24.                 X2 = (-b / denom) - (Math.Sqrt(disc) / denom);
25.                 Console.WriteLine("The Roots are .....: "+ X1 +" and "+X2);
26.             }
27.             else
28.             if (disc == 0)
29.             {
30.                 Console.WriteLine("The Roots are Repeated roots...");
31.                 X1 = -b / denom; Console.WriteLine("The Root is.....: " +X1);
32.             }
33.             else
34.             {
35.                 Console.WriteLine("The Roots are Imaginary roots...\n");
36.                 X1 = -b / denom; X2 = ((Math.Sqrt((4 * a * c) - (b * b))) / denom);
37.                 Console.WriteLine("The Root one.....: {0} +i{1}", X1, X2);
38.                 Console.WriteLine("The Roots are.....: {0} -i{1}", X1, X2);
39.             }
40.             Console.ReadLine();
41.         }
42.     }
43. }
```

.NET LAB PROGRAMS SOLUTION

4. Write a Program in C# to demonstrate boxing and unBoxing.

BoxingUnBoxing.cs

```
1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Text;
5.
6. namespace prog4
7. {
8.     class BoxingUnBoxing
9.     {
10.         static void Main(string[] args)
11.         {
12.             int num;
13.             Console.WriteLine("Enter the number:");
14.             num = int.Parse(Console.ReadLine());
15.             Object obj = num;
16.             Console.WriteLine("Value in num is : "+num);
17.             Console.WriteLine("Value in Object is : "+obj);
18.             int n;
19.             n = (int)obj;
20.             Console.WriteLine("Value in n is : "+n);
21.             Console.ReadLine();
22.         }
23.     }
24. }
```

5. Write a Program in C# to implement Stack operations.

Stack.cs

```
1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Text;
5.
6. namespace prog5
7. {
8.     class Stack
9.     {
10.         int[] stack = new int[10];
11.         int top = -1;
12.         void push(int data)
13.         {
14.             if (top == 10)
15.                 Console.WriteLine("Stack is full.");
16.             else
17.                 stack[++top] = data;
18.         }
19.         void pop()
20.         {
21.             if(top== -1)
22.                 Console.WriteLine("Stack is empty.");
23.             else
24.                 Console.WriteLine("Poped element is :"+stack[top--]);
25.         }
26.     }
27. }
```

.NET LAB PROGRAMS SOLUTION

```
25.     }
26.     void display()
27.     {
28.         if (top == -1)
29.             Console.WriteLine("Stack is empty.");
30.         else
31.             for(int temp=top;temp>=0;temp--)
32.                 Console.WriteLine(stack[temp]);
33.     }
34.     static void Main(string[] args)
35.     {
36.         int data, choice;
37.         Stack s = new Stack();
38.         do
39.         {
40.             Console.WriteLine("****STACK IMPLEMENTATION****");
41.             Console.WriteLine("1. PUSH\n2. POP.\n3. DISPLAY.\n4. EXIT.");
42.             Console.WriteLine("Enter Your Choice ?");
43.             choice = int.Parse(Console.ReadLine());
44.             switch (choice)
45.             {
46.                 case 1: Console.WriteLine("Enter element to insert :");
47.                     data = int.Parse(Console.ReadLine());
48.                     s.push(data);
49.                     break;
50.                 case 2: s.pop();
51.                     break;
52.                 case 3: s.display();
53.                     break;
54.                 case 4: Environment.Exit(0);
55.                     break;
56.                 default: Console.WriteLine("Invalid choice, Try again!!!");
57.                     break;
58.             }
59.         } while (true);
60.     }
61. }
62. }
```

6. Write a program to demonstrate Operator overloading.

Complex.cs

```
1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Text;
5.
6. namespace prog6
7. {
8.     class Complex
9.     {
10.         public int real;
11.         public int imaginary;
12.         public Complex(int real, int imaginary)
13.         {
14.             this.real = real;
15.             this.imaginary = imaginary;
```

.NET LAB PROGRAMS SOLUTION

```
16.         }
17.         public static Complex operator +(Complex c1, Complex c2)
18.         {
19.             return new Complex(c1.real + c2.real, c1.imaginary + c2.imaginary);
20.         }
21.         public override string ToString()
22.         {
23.             return (String.Format("{0} + {1}i", real, imaginary));
24.         }
25.         static void Main(string[] args)
26.         {
27.             Complex num1 = new Complex(2, 3);
28.             Complex num2 = new Complex(3, 4);
29.             Complex sum = num1 + num2;
30.             Console.WriteLine("First complex number : "+ num1);
31.             Console.WriteLine("Second complex number : "+ num2);
32.             Console.WriteLine("The sum of the two numbers : "+ sum);
33.             Console.ReadLine();
34.         }
35.     }
36. }
```

7. Write a Program in C# to find the second largest element in a single dimensional array.

SecondLargest.cs

```
1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Text;
5.
6. namespace prog7
7. {
8.     class SecondLargest
9.     {
10.         static void Main(string[] args)
11.         {
12.             int[] arr = new int[10];
13.             int n, i, big, sec_big;
14.             Console.WriteLine("How many element do you want to input ?");
15.             n=int.Parse(Console.ReadLine());
16.             Console.WriteLine("Enter "+n+" elements :");
17.             for(i=0;i<n;i++)
18.                 arr[i]=int.Parse(Console.ReadLine());
19.             if (arr[0] > arr[1])
20.             {
21.                 big = arr[0];
22.                 sec_big = arr[1];
23.             }
24.             else
25.             {
26.                 big = arr[1];
27.                 sec_big = arr[0];
28.             }
29.             for(i=2;i<n;i++)
30.             {
31.                 if (arr[i] > sec_big)
```

.NET LAB PROGRAMS SOLUTION

```
33.         if (arr[i] > big)
34.         {
35.             sec_big = big;
36.             big = arr[i];
37.         }
38.         else
39.             sec_big = arr[i];
40.     }
41.     Console.WriteLine("Second largest number is : " + sec_big);
42.     Console.ReadLine();
43. }
44. }
45. }
```

8. Write a Program in C# to multiply to matrices using Rectangular arrays.

MatrixMultiplication.cs

```
1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Text;
5.
6. namespace prog8
7. {
8.     class MatrixMultiplication
9.     {
10.         int[,] a;
11.         int[,] b;
12.         int[,] c;
13.         public void ReadMatrix()
14.         {
15.             Console.WriteLine("Enter the number of rows in Matrix 1 :");
16.             int m = int.Parse(Console.ReadLine());
17.             Console.Write("\nEnter the number of columns in Matrix 1 :");
18.             int n = int.Parse(Console.ReadLine());
19.             a = new int[m, n];
20.             Console.WriteLine("\nEnter the elements of Matrix 1:");
21.             for (int i = 0; i < a.GetLength(0); i++)
22.             {
23.                 for (int j = 0; j < a.GetLength(1); j++)
24.                 {
25.                     a[i, j] = int.Parse(Console.ReadLine());
26.                 }
27.             }
28.             Console.Write("\nEnter the number of rows in Matrix 2 :");
29.             m = int.Parse(Console.ReadLine());
30.             Console.Write("\nEnter the number of columns in Matrix 2 :");
31.             n = int.Parse(Console.ReadLine());
32.             b = new int[m, n];
33.             Console.WriteLine("\nEnter the elements of Matrix 2");
34.             for (int i = 0; i < b.GetLength(0); i++)
35.             {
36.                 for (int j = 0; j < b.GetLength(1); j++)
37.                 {
38.                     b[i, j] = int.Parse(Console.ReadLine());
39.                 }
40.             }
```

.NET LAB PROGRAMS SOLUTION

```
41.     }
42.     public void MultiplyMatrix()
43.     {
44.         if (a.GetLength(1) == b.GetLength(0))
45.         {
46.             c = new int[a.GetLength(0), b.GetLength(1)];
47.             for (int i = 0; i < c.GetLength(0); i++)
48.             {
49.                 for (int j = 0; j < c.GetLength(1); j++)
50.                 {
51.                     c[i, j] = 0;
52.                     for (int k = 0; k < a.GetLength(1); k++)
53.                         c[i, j] = c[i, j] + a[i, k] * b[k, j];
54.                 }
55.             }
56.         }
57.     }
58.     else
59.     {
60.         Console.WriteLine("\nNumber of columns in Matrix1 is not equal
to Number of rows in Matrix2.");
61.         Console.WriteLine("\nTherefore Multiplication of Matrix1 with
Matrix2 is not possible.");
62.         Console.ReadLine();
63.         Environment.Exit(-1);
64.     }
65. }
66. public void PrintMatrix()
67. {
68.     Console.WriteLine("\nGiven Matrix 1 is:");
69.     for (int i = 0; i < a.GetLength(0); i++)
70.     {
71.         for (int j = 0; j < a.GetLength(1); j++)
72.         {
73.             Console.Write(a[i, j] + "\t");
74.         }
75.         Console.WriteLine();
76.     }
77.     Console.WriteLine("\nGiven Matrix 2 is:");
78.     for (int i = 0; i < b.GetLength(0); i++)
79.     {
80.         for (int j = 0; j < b.GetLength(1); j++)
81.         {
82.             Console.Write(b[i, j] + "\t");
83.         }
84.         Console.WriteLine();
85.     }
86.     Console.WriteLine("\nMultiplying of Matrix 1 & Matrix 2 is:");
87.     for (int i = 0; i < c.GetLength(0); i++)
88.     {
89.         for (int j = 0; j < c.GetLength(1); j++)
90.         {
91.             Console.Write(c[i, j] + "\t");
92.         }
93.         Console.WriteLine();
94.     }
95. }
96. static void Main(string[] args)
```

.NET LAB PROGRAMS SOLUTION

```
97.     {
98.         MatrixMultiplication MatMul = new MatrixMultiplication();
99.         MatMul.ReadMatrix();
100.        MatMul.MultiplyMatrix();
101.        MatMul.PrintMatrix();
102.        Console.ReadLine();
103.    }
104. }
105. }
```

9. Find the sum of all the elements present in a jagged array of 3 inner arrays.

JagArraySum.cs

```
1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Text;
5.
6. namespace prog9
7. {
8.     class JagArraySum
9.     {
10.        static void Main(string[] args)
11.        {
12.            int[][] myJagArray = new int[3][];
13.            for (int i = 0; i < myJagArray.Length; i++)
14.            {
15.                myJagArray[i] = new int[i + 3];
16.            }
17.            for (int i = 0; i < 3; i++)
18.            {
19.                Console.WriteLine("Enter {1} elements of row {0} ", i,
myJagArray[i].Length);
20.                for (int j = 0; j < myJagArray[i].Length; j++)
21.                {
22.                    myJagArray[i][j] = int.Parse(Console.ReadLine());
23.                }
24.                Console.WriteLine();
25.            }
26.            int sum = 0;
27.            for (int i = 0; i < 3; i++)
28.            {
29.                for (int j = 0; j < myJagArray[i].Length; j++)
30.                {
31.                    sum += myJagArray[i][j];
32.                }
33.                Console.WriteLine("The sum of jagged array is : "+ sum);
34.                Console.ReadLine();
35.            }
36.        }
37.    }
38. }
```


.NET LAB PROGRAMS SOLUTION

10. Write a program to reverse a given string using C#.

StringReverse.cs

```
1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Text;
5.
6. namespace prog10
7. {
8.     class StringReverse
9.     {
10.         static void Main(string[] args)
11.         {
12.             string str;
13.             Console.WriteLine("Enter a string:");
14.             str = Console.ReadLine();
15.             int len = str.Length;
16.             char[] revstr = new char[len];
17.             for (int i = 0; i < len; i++)
18.             {
19.                 revstr[i] = str[len - 1 - i];
20.             }
21.             Console.WriteLine("The reverse of given string is : "+new string(revstr));
22.             Console.ReadLine();
23.         }
24.     }
25. }
```

11. Using Try, Catch and Finally blocks write a program in C# to demonstrate error handling.

ExceptionHandling.cs

```
1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Text;
5.
6. namespace prog11
7. {
8.     class ExceptionHandling
9.     {
10.         public static int divide(int a, int b)
11.         {
12.             int d=-1;
13.             try
14.             {
15.                 Console.WriteLine("try block.\n");
16.                 d = a / b;
17.             }
18.             catch (InvalidCastException e)
19.             {
20.                 Console.WriteLine("catch block1.\n");
21.                 Console.WriteLine(e);
22.             }
23.             catch (DivideByZeroException e)
24.             {
```

.NET LAB PROGRAMS SOLUTION

```
25.         Console.WriteLine("catch block2.\n");
26.         Console.WriteLine(e);
27.     }
28.     finally
29.     {
30.         Console.WriteLine("finally block.");
31.     }
32.     return d;
33. }
34.
35. static void Main(string[] args)
36. {
37.     int a, b;
38.     Console.WriteLine("Enter two number:");
39.     a = int.Parse(Console.ReadLine());
40.     b = int.Parse(Console.ReadLine());
41.     Console.WriteLine("Result of a/b is : "+ devide(a,b));
42.     Console.ReadLine();
43. }
44. }
45. }
```

12. Design a simple calculator using Switch Statement in C#.

Calculator.cs

```
1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Text;
5.
6. namespace prog12
7. {
8.     class Calculator
9.     {
10.         static void Main(string[] args)
11.         {
12.             double a, b;
13.             int choice;
14.             do
15.             {
16.                 Console.WriteLine("Select the operation:");
17.                 Console.WriteLine("1: Addition");
18.                 Console.WriteLine("2: Subtraction");
19.                 Console.WriteLine("3: Multiplication");
20.                 Console.WriteLine("4: Division");
21.                 Console.WriteLine("0: Exit");
22.                 Console.WriteLine("Enter your choice:");
23.                 choice = int.Parse(Console.ReadLine());
24.                 switch (choice)
25.                 {
26.                     case 1: Console.WriteLine("Enter the two numbers:");
27.                         a = double.Parse(Console.ReadLine());
28.                         b = double.Parse(Console.ReadLine());
29.                         Console.WriteLine("The result of addition is:" + (a + b));
30.                         break;
31.                     case 2: Console.WriteLine("Enter the two numbers:");
```

.NET LAB PROGRAMS SOLUTION

```
32.         a = double.Parse(Console.ReadLine());
33.         b = double.Parse(Console.ReadLine());
34.         Console.WriteLine("The result of subtraction is:" + (a - b));
35.         break;
36.     case 3: Console.WriteLine("Enter the two numbers:");
37.         a = double.Parse(Console.ReadLine());
38.         b = double.Parse(Console.ReadLine());
39.         Console.WriteLine("The result of multiplication is:" + (a * b));
40.         break;
41.     case 4: Console.WriteLine("Enter the two numbers:");
42.         a = double.Parse(Console.ReadLine());
43.         b = double.Parse(Console.ReadLine());
44.         if (b == 0)
45.             Console.WriteLine("Division is not possible.");
46.         else
47.             Console.WriteLine("The result of division is:" + (a / b));
48.         break;
49.     case 0: Environment.Exit(-1);
50.         break;
51.     default: Console.WriteLine("Invalid choice, Try again.\n");
52.         break;
53.     }
54. } while (true);
55. }
56. }
57. }
```

13. Demonstrate Use of Virtual and override key words in C# with a simple program

MethodOverriding.cs

```
1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Text;
5.
6. namespace prog13
7. {
8.     public class Customer
9.     {
10.         public virtual void CustomerType()
11.         {
12.             Console.WriteLine("I am a Customer");
13.         }
14.     }
15.     public class CorporateCustomer : Customer
16.     {
17.         public override void CustomerType()
18.         {
19.             Console.WriteLine("I am a Corporate Customer");
20.         }
21.     }
22.     public class PersonalCustomer : Customer
23.     {
24.         public override void CustomerType()
25.         {
26.             Console.WriteLine("I am a Personal Customer");
27.         }
28.     }
29. }
```

.NET LAB PROGRAMS SOLUTION

```
28.     }
29.
30.     class MethodOverriding
31.     {
32.         static void Main(string[] args)
33.         {
34.             Customer[] c = new Customer[3];
35.             c[0] = new PersonalCustomer();
36.             c[1] = new CorporateCustomer();
37.             c[2] = new Customer();
38.             foreach (Customer CustomerObject in c)
39.             {
40.                 CustomerObject.CustomerType();
41.             }
42.             Console.ReadLine();
43.         }
44.     }
45. }
```

14. Implement linked lists in C# using the existing collections name space.

MyList.cs

```
1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Text;
5.
6. namespace prog14
7. {
8.     class MyList
9.     {
10.         static void Main(string[] args)
11.         {
12.             LinkedList<int> list = new LinkedList<int>();
13.             list.AddFirst(10);
14.             list.AddLast(50);
15.             Console.WriteLine("The elements in the linked list are:");
16.             foreach (int i in list)
17.                 Console.WriteLine(i);
18.             list.RemoveFirst();
19.             Console.WriteLine("Elements in the linked list after deleting are:");
20.             foreach (int i in list)
21.                 Console.WriteLine(i);
22.             Console.ReadLine();
23.         }
24.     }
25. }
```

15. Write a program to demonstrate abstract class and abstract methods in C#.

AbstractClassAndMethod.cs

```
1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Text;
5.
```

.NET LAB PROGRAMS SOLUTION

```
6. namespace prog15
7. {
8.     abstract class Shape
9.     {
10.         abstract public void show();
11.     }
12.     class Circle : Shape
13.     {
14.         public override void show()
15.         {
16.             Console.WriteLine("We are in Circle");
17.         }
18.     }
19.     class Triangle : Shape
20.     {
21.         public override void show()
22.         {
23.             Console.WriteLine("We are in Triangle");
24.         }
25.     }
26.     class AbstractClassAndMethod
27.     {
28.         static void Main(string[] args)
29.         {
30.             Circle c = new Circle();
31.             Triangle t = new Triangle();
32.             c.show();
33.             t.show();
34.             Console.ReadLine();
35.         }
36.     }
37. }
```

16. Write a program in C# to build a class which implements an interface which is already existing.

InterfaceExample.cs

```
1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Text;
5.
6. namespace prog16
7. {
8.     interface Shape
9.     {
10.         void Show();
11.     }
12.     class Triangle:Shape
13.     {
14.         #region Shape Members
15.         void Shape.Show()
16.         {
17.             Console.WriteLine("I am printing from a Triangle");
18.         }
19.         #endregion
20.     }
21.     class Circle:Shape
```

.NET LAB PROGRAMS SOLUTION

```
22.     {
23.         #region Shape Members
24.         void Shape.Show()
25.         {
26.             Console.WriteLine("I am printing from Circle");
27.         }
28.         #endregion
29.     }
30.
31.     class InterfaceExample
32.     {
33.         static void Main(string[] args)
34.         {
35.             Shape s;
36.             Triangle t = new Triangle();
37.             Circle c = new Circle();
38.             s = c;
39.             s.Show();
40.             s = t;
41.             s.Show();
42.             Console.ReadLine();
43.         }
44.     }
45. }
```

17. Write a program to illustrate the use of different properties in C#.

Student.cs

```
1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Text;
5.
6. namespace prog17
7. {
8.     class Student
9.     {
10.         private string name, course;
11.         private int marks;
12.         // Declare a Name property of type string:
13.         public string Name
14.         {
15.             get
16.             {
17.                 return name;
18.             }
19.             set
20.             {
21.                 name = value;
22.             }
23.         }
24.
25.         // Declare a course property of type string:
26.         public string Course
27.         {
28.             get
29.             {
```

.NET LAB PROGRAMS SOLUTION

```
30.         return course;
31.     }
32.     set
33.     {
34.         course = value;
35.     }
36. }
37. // Declare a Marks property of type int:
38. public int Marks
39. {
40.     get
41.     {
42.         return marks;
43.     }
44.     set
45.     {
46.         marks = value;
47.     }
48. }
49. public override string ToString()
50. {
51.     return "\nName = " + Name + "\nCourse = " + Course + "\nMarks = " + Marks;
52. }
53. static void Main(string[] args)
54. {
55.     // Create a new Student object:
56.     Student s = new Student();
57.     // Setting name, course and marks of the student
58.     s.Name = "ABC";
59.     s.Course = "MCA";
60.     s.Marks = 835;
61.     Console.WriteLine("Student Details : "+ s);
62.     Console.ReadLine();
63. }
64. }
65. }
```

18. Demonstrate arrays of interface types with a C# program.

StudentInfo.cs

```
1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Text;
5.
6. namespace prog18
7. {
8.     interface Inter
9.     {
10.         void info();
11.     }
12.     class Student : Inter
13.     {
14.         private string name, course;
15.         private int marks;
16.         public Student(string name, string course, int marks)
17.         {
```

.NET LAB PROGRAMS SOLUTION

```
18.         this.name = name;
19.         this.course = course;
20.         this.marks = marks;
21.     }
22.     public void info()
23.     {
24.         Console.WriteLine(name + "\t" + course + "\t" + marks);
25.     }
26. }
27. class StudentInfo
28. {
29.     static void Main(string[] args)
30.     {
31.         Inter[] obj = { new Student("abc", "mca", 590), new Student("def",
"bca", 650), new Student("ghi", "mba", 855) };
32.         Console.WriteLine("NAME\tCOURSE\tMARKS\n-----
");
33.         foreach (Inter s in obj)
34.             s.info();
35.         Console.ReadLine();
36.     }
37. }
38. }
```

Note: In the examination *each* student picks one question from the lot of *all* 18 questions.