

Student Record System

Matriculation Number:

First Name:

Second Name:

Coursework Mark:

Exam Mark:

Date of Birth:

Stored Students:

Student Record System

Matriculation Number:

First Name:

Second Name:

Coursework Mark:

Exam Mark:

Date of Birth:

Stored Students:

Student Record System

Matriculation Number:

First Name:

Second Name:

Coursework Mark:

Exam Mark:

Date of Birth:

Stored Students:

10001

classList

```

1 // Author: Christopher Johnson [40275286]
2 // Class Purpose: Student C#ps to be created within the main window
3 // Date Last Modified: 23/10/2017
4
5 using System;
6 using System.Collections.Generic;
7 using System.Linq;
8 using System.Text;
9
10 namespace BusinessObjects
11 {
12     13 references
13     public class Student
14     {
15         // Declaring the attributes for the Student Class
16         private int _matricNo;
17         private string _firstName;
18         private string _secondName;
19         private int _courseMark;
20         private int _examMark;
21         private DateTime _dateOfBirth;
22
23         5 references
24         public int Matric
25         {
26             get
27             {
28                 return _matricNo;
29             }
30             set
31             {
32                 // Validates that the matric number provided is between the accepted values
33                 if(value < 10001 || value > 50000)
34                 {
35                     // Generates an error which is passed back to the calling method
36                     throw new ArgumentException("Matriculation Number must be between 10001 and 50000");
37                 }
38                 _matricNo = value;
39             }
40         }
41         public string FirstName
42         {
43             get
44             {
45                 return _firstName;
46             }
47             set
48             {
49                 // Validates that the First name provided isn't blank or whitespace
50                 if(string.IsNullOrEmpty(value))
51                 {
52                     // Generates error if if statement is true
53                     throw new ArgumentException("First Name cannot be blank");
54                 }
55                 _firstName = value;
56             }
57         }
58         3 references
59         public string SecondName
60         {
61             get
62             {
63                 return _secondName;
64             }
65             set
66             {
67                 // Validates that the Second Name provided isn't blank or whitespace
68                 if (string.IsNullOrEmpty(value))
69                 {
70                     throw new ArgumentException("Second Name cannot be blank");
71                 }
72                 _secondName = value;
73             }
74         }
75         3 references
76         public int courseworkMark
77         {
78             get
79             {
80                 return _courseMark;
81             }
82             set
83             {
84                 // Validates that the coursework mark provided is within the accepted range
85                 if (value > 20 || value < 0 )
86                 {
87                     throw new ArgumentException("Coursework Mark must be out of 20");
88                 }
89                 _courseMark = value;
90             }
91         }
92         3 references
93         public int examMark
94         {
95             get
96             {
97                 return _examMark;
98             }
99             set
100             {
101                 // Validates that the coursework mark provided is within the accepted range
102                 if (value > 40 || value < 0 )
103                 {
104                     throw new ArgumentException("Exam Mark must be out of 40");
105                 }
106                 _examMark = value;
107             }
108         }
109         4 references
110         public DateTime dateOfBirth
111         {
112             get
113             {
114                 return _dateOfBirth;
115             }
116             set
117             {
118                 // Validates that the date of birth proved is not blank or white space
119                 if(dateOfBirth == null)
120                 {
121                     throw new ArgumentException("Date of Birth cannot be blank");
122                 }
123                 _dateOfBirth = value;
124             }
125         }
126     }
127 }

```