```
// q1
namespace Week9
{
    class Utility
    {
        /*
        public static uint GetUInt(string message)
        {
            uint data = 0;
            try
            {
                Console.Write(message);
                data = uint.Parse(Console.ReadLine());
            }
            catch(Exception e)
            { Console.WriteLine("Invalid input {0} - try again", e.Message); }
            return data;
        }
        */
        public static uint GetUInt(string message)
        {
            uint data = 0;
            Console.Write(message);
            data = uint.Parse(Console.ReadLine());

            return data;
        }

        public static string GetString(string message)
        {
            string data = "";
            while(data == null || data == "")
            {
                Console.Write(message);
                data = Console.ReadLine();
            }
            return data;
        }
    }
}

namespace Week9
{
    class Unit
    {

        // private instance variables/attributes
        private string _UnitCode, _UnitName;
        private double _Mark;
```

```
public string UnitCode { get { return _UnitCode; } }
public string UnitName { get { return _UnitName; } }
public double Mark { get { return _Mark; } }
public string Grade
{
    get
    {
        string grade = "";
        if (_Mark < 50) grade = "N";
        else if (_Mark < 60) grade = "P";
        else if (_Mark < 70) grade = "C";
        else if (_Mark < 80) grade = "D";
        else grade = "HD";
        return grade;
    }
}

public Unit()
{
    string name = "";
    while (name == "")
        try
        {
            name = Utility.GetString("Enter Unit Code: ");
            if (name == "") throw new FormatException("Unit Code can't be blank");
            else if (name.Length != 6 || name.Contains(" "))
                throw new FormatException("Unit Code must be 6 characters in one word");
            else
            {
                int i;
                for (i = 0; i < 3; i++)
                    if (Char.IsLetter(name[i]) == false)
                        throw new FormatException("Unit Code must start with 3 characters");
                for (; i < 6; i++)
                    if (Char.IsDigit(name[i]) == false)
                        throw new FormatException("Unit Code must End with 3 digits");
                if (i == 6) _UnitCode = name.ToUpper();
            }
        }
        catch (FormatException e)
        {
            Console.WriteLine("Invalid {0}", e.Message);
            name = "";
        }
```

```
            name = "";
            while (name == "")
             try
             {
                name = Utility.GetString("Enter Unit Name: ");
                if (name == "") throw new FormatException("Unit Name can't be blank");
                else
                {
                   int i;
                   for (i = 0; i < name.Length; i++)
                      if (Char.IsDigit(name[i]))
                         throw new FormatException("Unit Name must be characters");
                   if (i == name.Length) _UnitName = name.ToUpper();
                }
             }
             catch (FormatException e)
             {
                Console.WriteLine("Invalid {0}", e.Message);
                name = "";
             }
           uint m = 0;
           while (m == 0)
             try
             {
                m = Utility.GetUInt("Enter Mark: ");
                if (m > 100) throw new FormatException("Please check the Mark input 0-100 Only");
                else _Mark = Convert.ToDouble(m);
             }
             catch (Exception e)
             {
                Console.WriteLine("Invalid {0}", e.Message);
                m = 0;
             }
       }
       public override string ToString()
       {
          return string.Format("{0}: {1,-25} {2,6:f} {3,2}", _UnitCode, _UnitName, _Mark, Grade);
       }

    }
}
namespace Week9
{
    class Student
    {
       // instance variables/attributes
       private string _ID, _Name, _Phone;
       private DateTime _DOB;
       private List<Unit> _UnitList;
```

```
// ReadOnly properties for all attributes
public string ID
{
    get { return _ID; }
}
public string Name
{
    get { return _Name; }
}
public DateTime DOB {  get { return _DOB;  } }
public string Phone
{
    get { return _Phone; }
}
public ReadOnlyCollection<Unit> UnitList
{
    get { return _UnitList.AsReadOnly(); }
}
// constructor
public Student() // parameter-less to set data with default value
{
    // _ID = "2011000000";
    // _FirstName = "No First Name";
    // _LastName = "No Surname";
    uint id = 0;
    //while (id == 0 || id.ToString().Length < 8 || id.ToString().Length > 9)
    //    id = Utility.GetUInt("Enter Student ID: ");
    while (id == 0)
        try
        {
            id = Utility.GetUInt("Enter Student ID: ");
            if (id.ToString().Length < 8 || id.ToString().Length > 9)
                throw new FormatException("ID MUST be 8 or 9 Digits");
            _ID = id.ToString();
        }
        catch (FormatException e)
        {
            Console.WriteLine("Invalid {0}", e.Message);
            id = 0;
        }
    string name = "";
    while (name == "")
        try
        {
            name = Utility.GetString("Enter Student Full Name: ");
            if (name == "") throw new FormatException("Name can't be blank");
            else if (name.Contains(" ") == false || name.StartsWith(" ") || name.EndsWith(" "))
                throw new FormatException("Must have First and Last name");
            _Name = name;
        }
```

```
    catch (FormatException e)
    {
        Console.WriteLine("Invalid {0}", e.Message);
        name = "";
    }

uint y = 0;
while (y == 0)
    try
    {
        y = Utility.GetUInt("Enter Student's Year of Birth: ");
        if (y.ToString().Length != 4)
            throw new FormatException("Year must be in YYYY format");
        else if (y < DateTime.Now.Year - 100)
            throw new FormatException("SORRY! too old to attend the class");
        else if (y > DateTime.Now.Year - 17)
            throw new FormatException("You are too young to study at Uni");
    }
    catch (FormatException e)
    {
        Console.WriteLine("Invalid {0}", e.Message);
        y = 0;
    }

uint m = 0;
while (m == 0)
    try
    {
        m = Utility.GetUInt("Enter Month of Birth: ");
        if (m < 1 || m > 12) throw new FormatException("Please check the Month input");
    }
    catch (Exception e)
    {
        Console.WriteLine("Invalid {0}", e.Message);
        m = 0;
    }
DateTime dob;
uint d = 0;
while (d == 0)
    try
    {
        d = Utility.GetUInt("Enter Date of Birth: ");
        dob = new DateTime(Convert.ToInt32(y), Convert.ToInt32(m), Convert.ToInt32(d));
        _DOB = dob;
    }
    catch (Exception e)
    {
        Console.WriteLine("Invalid {0}", e.Message);
        d = 0;
    }
```

```
            id = 0;
            while (id == 0)
                try
                {
                    id = Utility.GetUInt("Enter Student Mobile Phone: ");
                    if (id.ToString().Length != 9)
                        throw new FormatException("Mobile phone must be 9 digits");
                    _Phone = id.ToString();
                }
                catch (FormatException e)
                {
                    Console.WriteLine("Invalid {0}", e.Message);
                    id = 0;
                }
            _UnitList = new List<Unit>();
            for(int i = 0; i < 4; i++)
            {
                Unit u = new Unit();
                _UnitList.Add(u);
            }
        }

        // No custom or parameterised constructor - set the private with appropriate given parameter

        // NOcopy constructor - assign all the properties from parameter to the private variables

        public override string ToString()
        {
            string temp = string.Format("{0} {1} - Mobile Phone: {2} - Date Of Birth: {3:d2}/{4:d2}/{5:d4}",
                        _ID, _Name, _Phone, _DOB.Day, _DOB.Month, _DOB.Year);
            foreach (Unit u in _UnitList) temp += "\n\t- " + u.ToString();
            return temp + "\n\n";
        }
    }
}
namespace Week9
// Task1 get Student ID, name, Age(DOB), Phone, 4 unit results Unit code, Name, marks
    class Program
    {
        static void Main(string[] args)
        {

            Student demo = new Student();
            Console.WriteLine(demo);

        }
    }
}
```

Q2

```
class Deck
 {
 .......
      public Card Deal() // THIS FUNCTION MODIFIED FOR PART (a)
      {
         Card c = null;
         int which_suit;
         while (c == null)
         {
            which_suit = _Random.Next(1, Deck.MAX_SUITS);
            try {  c = _Suits[which_suit - 1].Deal();   }
            catch (Exception e)  {   Console.Error.WriteLine("Exception occurred: {0}", e.Message);  }
         }
         return c;
      }
 }

   class Program
   {
      public static void Main()  // THIS FUNCTION MODIFIED FOR PART (b)
      {
         Dealer dealer = new Dealer();
         Hand theHand;
         char keepGoing;
         do
         {
            try
            {   theHand = dealer.Deal();
               while (true)
               {
                  Console.WriteLine("The hand is as follows:");
                  theHand.Print();
                  Console.WriteLine("The hand is worth {0}", theHand.GetValue());
                  Console.WriteLine();
                  Console.WriteLine("Press ENTER to deal another card...");
                  Console.ReadLine();
                  dealer.Hit();
               }
            }
            catch (Exception e)    {   Console.Error.WriteLine("An exception occurred: {0}", e.Message);   }
            Console.WriteLine();
            Console.Write("Do you wish to try again (Y/N)? ");
            keepGoing = char.ToUpper((Console.ReadLine()[0]));
         } while (keepGoing == 'Y');
      }
   }
```