# C# .Net Programming Assignment 11

- **Create separate visual Studio project for each problem statement separately.**
- **For Business logic write separate class.**
- **Use Object Oriented concepts while writing the program.**

1. **Write a program which accept password from user. Password should contains at least 3 capital, 2 small and 2 digits in it. Minimum length of password should be 7. If any of the given condition is not satisfied by the user then we have to throw user defined exceptions named as InvalidPasswordException.**

```
using System;

public class InvalidPasswordException : Exception
{
            // Logic (At least one parametrised constructor)
}

public class MarvellousAuthentication
{
        public string password;

        public void MarvellousAuthentication(string str)
        {
                password = ____________ ;
        }

        public boolean ChkPassword()
        {
                // Logic to check password
                // return true or false depends on contents
        }
}

class Marvellous
{
        static void Main(string[] args)
        {
                MarvellousAuthentication mobj;
                boolean bret ;

                Console.WriteLine("Enter password");
                string str = Console.ReadLine();
```

```
                    mobj = new MarvellousAuthentication(str);

                    bret = mobj.ChkPassword(str);

                    try
                    {
                            if(bret == ________ )
                            {
                                    // throw user defined exception
                            }
                            else
                            {
                                    // Print success message
                            }
                    }

                    catch(____________)
                    {
                            // Logic
                    }
            }
    }
```

2.**Write a program which contains one class named as MarvellousEmployee which holds information of employee.We have to accept all the information of employee from user and depends on the provided information we have to throw user defined exception.**

**If name of employee contains other than character then throw InvalidNameException. If age of employee is negative then throw AgeInvalidException. If blood group of employee is there than O,A,B,AB then throw BloodGroupInvalidException.**

**All the exceptions are considered as user defined exception.**

```
    using System;

    public class InvalidNameException : Exception
    {
            // Logic (At least one parametrised constructor)
    }

    public class AgeInvalidException : Exception
    {
            // Logic (At least one parametrised constructor)
    }
```

```
public class BloodGroupInvalidException : Exception
{
            // Logic (At least one parametrised constructor)
}

public class MarvellousEmployee
{
        public string EName;
        public int EAge;
        public string EBloodGroup;

        public void Accept()
        {
                // Accept name, blood group, and age from user
                // Depends on validations throw specific exception
        }

        public void Display()
        {
                // Logic to display all details
        }
}

class Marvellous
{
        static void Main(string[] args)
        {
                MarvellousEmployee mobj = new MarvellousEmployee();

                try
                {
                        mobj.Accept();
                }

                catch(____________)
                {
                        // Logic
                }

                catch(____________)
                {
                        // Logic
                }
                catch(____________)
                {
                        // Logic
                }
                mobj.Display();
        }
}
```