Ans To The Qus. No -1

(a)

Entities:

1. Doctor
2. Department
3. Service Point

Attributes:

1. Doctor ID (primary key)
2. Doctor Name
3. Contact Number
4. Department ID (foreign key)
5. Department Name
6. Service Point ID (foreign key)
7. Service Point Name

Based on these entities and attributes, we can apply normalization rules as follows:

Doctor Table:

- Doctor ID (primary key)
- Doctor Name
- Contact Number

Department Table:

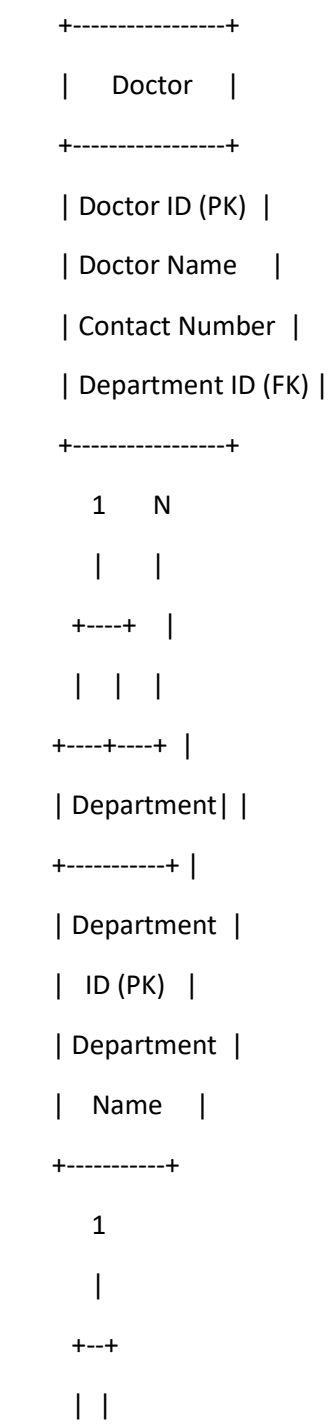
- Department ID (primary key)
- Department Name

Service Point Table:

- Service Point ID (primary key)
- Service Point Name

(b)

Now, let's draw the Entity Relationship Diagram (ERD) with crow's foot notation for the normalized tables:



+-----+

|Service|

| Point |

+-------+

|Service|

|Point ID|

| (PK) |

|Service|

| Point |

| Name |

+-------+

Ans To The Qus. No -2

```
int n = 30;
for (int i = 0; i <= 5; i++)
{
   n += i;
   Console.WriteLine(n);
}
```

Output:

30

31

33

36

40

45

Ans To The Qus. No -4

Method Overloading:

Method overloading is a feature in object-oriented programming where multiple methods can have the same name but differ in their parameters. In C#, method overloading allows us to define multiple methods with the same name but with different parameter lists.

Example of Method Overloading in C#:

```
using System;

class Calculator
{
    public int Add(int number1, int number2)
    {
        return number1 + number2;
    }

    public int Add(int number1, int number2, int number3)
    {
        return number1 + number2 + number3;
    }
}

class Program
{
    static void Main(string[] args)
    {
        Calculator calculator = new Calculator();

        int sum1 = calculator.Add(2, 3);
```

```
        Console.WriteLine("Sum1: " + sum1);


        int sum2 = calculator.Add(2, 3, 4);
        Console.WriteLine("Sum2: " + sum2);
    }
}
```

Method Overriding:

Method overriding is a feature in object-oriented programming that allows a subclass to provide a specific implementation of a method that is already defined in its superclass.

Example of Method Overriding in C#:

```
using System;

class Shape
{
    public virtual void Draw()
    {
        Console.WriteLine("Drawing a shape.");
    }
}

class Circle : Shape
{
    public override void Draw()
    {
        Console.WriteLine("Drawing a circle.");
    }
```

```
}

class Rectangle : Shape
{
    public override void Draw()
    {
        Console.WriteLine("Drawing a rectangle.");
    }
}

class Program
{
    static void Main(string[] args)
    {
        Shape shape1 = new Circle();
        shape1.Draw();

        Shape shape2 = new Rectangle();
        shape2.Draw();
    }
}
```

Ans To The Qus. No -5

```
using System;

public abstract class Clinician
{
    // Properties
```

```
    public string Name { get; set; }
    public string HospitalName { get; set; }

    // Constructor
    public Clinician(string name, string hospitalName)
    {
        Name = name;
        HospitalName = hospitalName;
    }

    // Methods
    public bool Login(string userName, string password)
    {
        if (IsSessionExists(userName))
        {
            Console.WriteLine("Session already exists. Please logout before logging in again.");
            return false;
        }

        // Perform actual login logic here
        Console.WriteLine("Clinician login successful.");
        return true;
    }

    private bool IsSessionExists(string userName)
    {
            return false;
    }
}
```

```
public class Doctor : Clinician
{
    // Properties
    public string PracticeNumber { get; set; }

    // Constructor
    public Doctor(string name, string hospitalName, string practiceNumber)
        : base(name, hospitalName)
    {
        PracticeNumber = practiceNumber;
    }

    // Methods
    public void CreatePrescription(int patientNumber)
    {
        Console.WriteLine($"{Name} is creating a prescription for patient number {patientNumber}.");
    }
}

public class Pharmacist : Clinician
{
    // Properties
    public string PharmacistNumber { get; set; }

    // Constructor
    public Pharmacist(string name, string hospitalName, string pharmacistNumber)
        : base(name, hospitalName)
    {
```

```
        PharmacistNumber = pharmacistNumber;
    }

    // Methods
    public void DispenseMedications(int prescriptionNumber)
    {
        Console.WriteLine($"{Name} is dispensing medications for prescription number
{prescriptionNumber}.");
    }
}
```

Ans To The Qus. No -6

```
using System;

class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Enter three integer values:");
        int n1 = int.Parse(Console.ReadLine());
        int n2 = int.Parse(Console.ReadLine());
        int n3 = int.Parse(Console.ReadLine());

        int min;

        if (n1 < n2)
        {
            min = n1;
        }
```

```
        else
        {
            min = n2;
        }


        if (n3 < min)
        {
            min = n3;
        }

        Console.WriteLine("The minimum value is: " + min);
    }
}
```