Object Oriented Programming
[Understanding the Concept of Overloading]

Task No. 1: Write a program which contains a class 'Calculator' contains multiple sum method by using method overloading concept.

Solution:

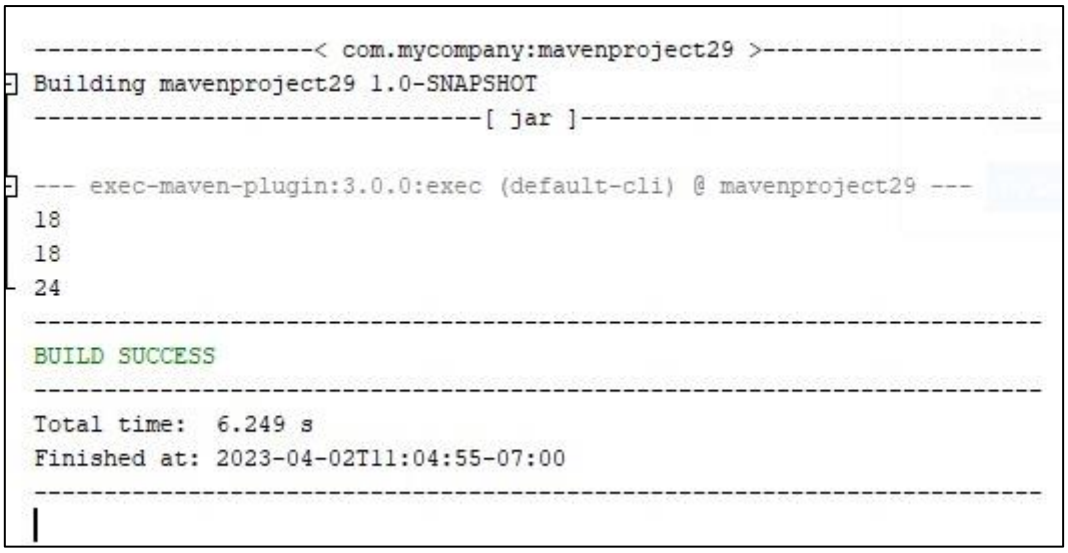
Main Method:

```
package com.mycompany.mavenproject29;

public class Mavenproject29 {

    public static void main(String[] args) {

        Calculator cal = new Calculator();

        System.out.println(cal.sum(10,8));

        System.out.println(cal.sum(4,6,8));

      System.out.println( cal.sum(7,5,3,9));

    }

}
```

Class:

```
package com.mycompany.mavenproject29;

public class Calculator {

      public  int sum(int num1, int num2){

      return num1+num2;

   }

     public  int sum(int num1, int num2,int num3){

      return num1+num2+num3;

   }

     public  int sum(int num1, int num2,int num3, int num4){

    return num1+num2+num3+num4;

   }

}
```

Output:



```
--------------------< com.mycompany:mavenproject29 >--------------------
Building mavenproject29 1.0-SNAPSHOT
--------------------------------[ jar ]---------------------------------

--- exec-maven-plugin:3.0.0:exec (default-cli) @ mavenproject29 ---
18
18
24
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  6.249 s
Finished at: 2023-04-02T11:04:55-07:00
------------------------------------------------------------------------
```

Task No. 2: Create a class to print the area of a square and a rectangle. The class has two methods with the same name but different number of parameters. The method for printing area of rectangle has two parameters which are length and breadth respectively while the other method for printing area of square has one parameter which is side of square.

Solution:

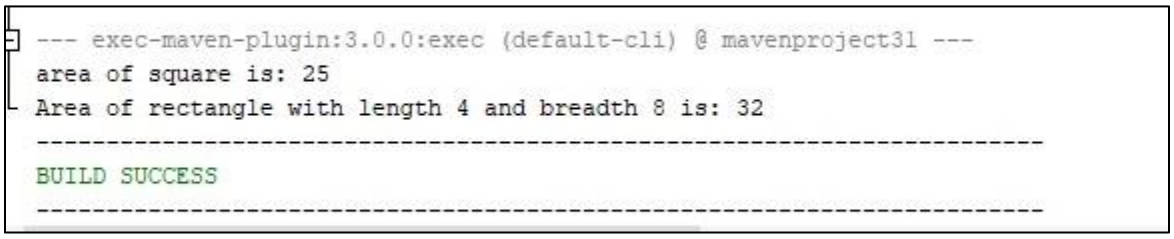
Main Method:

```
package com.mycompany.mavenproject31;

public class Mavenproject31 {

   public static void main(String[] args) {

      java shape = new java();

      shape.area(5);

      shape.area(4, 8);

   }

}
```

Class:

```
package com.mycompany.mavenproject31;

public class java {

     public void area(int side){

    int area =side*side;

     System.out.println("area of square is: "+ area);

   }

       public void area(int length, int breadth){

    int area =  length*breadth;

     System.out.println("Area of rectangle with length " + length + " and breadth " + breadth + " is: " + 
area);


   }

}
```

Output:



```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ mavenproject31 ---
area of square is: 25
Area of rectangle with length 4 and breadth 8 is: 32
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
```

Task No. 3: Create a class 'Student' with three data members which are name, age and address. The constructor of the class assigns default values name as "unknown", age as '0' and address as "not available". It has two members with the same name 'setInfo'. First method has two parameters for name and age and assigns the same whereas the second method takes has three parameters which are assigned to name, age and address respectively. Print the name, age and address of 4 students.

Solution:

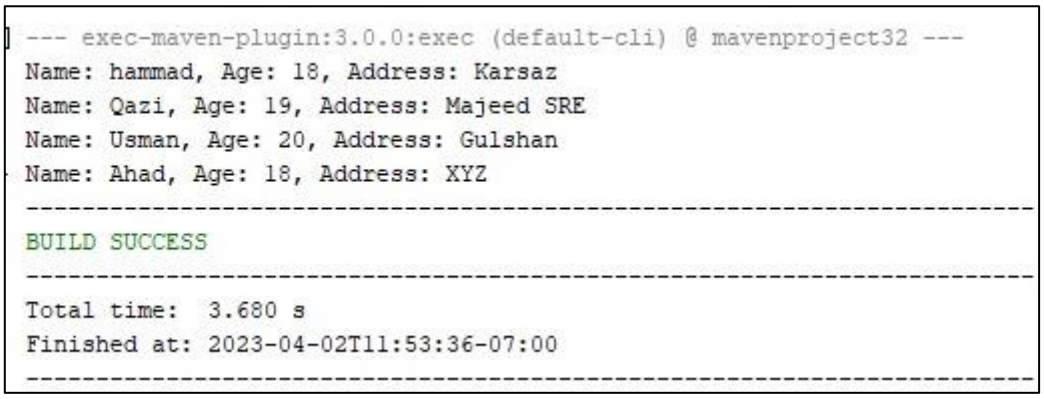
Main Method:

```
package com.mycompany.mavenproject32;
public class Mavenproject32 {

   public static void main(String[] args) {

       student s = new student();
      s.setInfo("hammad",18,"Karsaz");
      s.printInfo();

        student s1 = new student();
      s1.setInfo("Qazi",19,"Majeed SRE");
      s1.printInfo();

        student s2 = new student();
      s2.setInfo("Usman",20,"Gulshan");
      s2.printInfo();

        student s3 = new student();
      s3.setInfo("Ahad",18,"XYZ");
      s3.printInfo();
   }
}
```

Class:

```
package com.mycompany.mavenproject32;
public class student {

    private String name;
    private int age;
    private String address;

    public student() {
        this.name = "unknown";
        this.age = 0;
        this.address = "not available";
    }

     public void setInfo(String name, int age) {
        this.name = name;
        this.age = age;
    }

      public void setInfo(String name, int age, String address) {
        this.name = name;
        this.age = age;
        this.address = address;
    }

       public void printInfo() {
        System.out.println("Name: " + this.name + ", Age: " + this.age + ", Address: " + this.address);
    }
}
```

Output:



```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ mavenproject32 ---
Name: hammad, Age: 18, Address: Karsaz
Name: Qazi, Age: 19, Address: Majeed SRE
Name: Usman, Age: 20, Address: Gulshan
Name: Ahad, Age: 18, Address: XYZ
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  3.680 s
Finished at: 2023-04-02T11:53:36-07:00
------------------------------------------------------------------------
```

Task No. 4: Implement the Circle class to overload the + operator so that you can add two Circle objects. Adding two Circle object should give another Circle whose radius is the sum of the radii of the two Circle objects.

Solution:

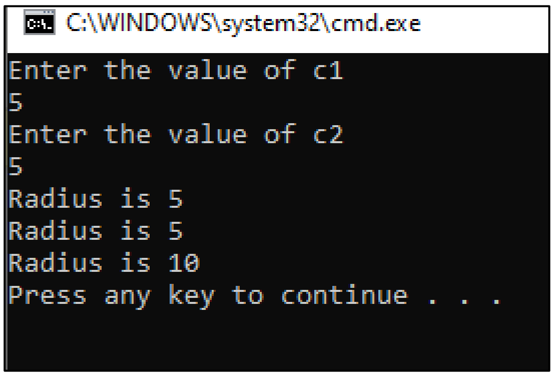
Main Method:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApp17
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Enter the value of c1");
            int n = int.Parse(Console.ReadLine());

            Console.WriteLine("Enter the value of c2");
            int y = int.Parse(Console.ReadLine());

            Circle c1 = new Circle(n);
            Circle c2 = new Circle(y);
            Circle c3 = c1 + c2;
            c1.display();
            c2.display();
            c3.display();

        }
    }
}
```

Class:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApp17
{
    class Circle
    {
        int radius;
        public Circle()
        {
            this.radius = 0;
         }
        public Circle(int radius)
        {
            this.radius = radius;
        }
        public void display()
        {
            Console.WriteLine("Radius is " + radius);
        }
        public static Circle operator +(Circle c1, Circle c2)
        {
            Circle c3 = new Circle();
            c3.radius = c1.radius + c2.radius;
            return c3;
        }

    }
}
```

Output:


C:\WINDOWS\system32\cmd.exe

```
Enter the value of c1
5
Enter the value of c2
5
Radius is 5
Radius is 5
Radius is 10
Press any key to continue . . .
```

Task No. 5: Implement the Rectangle class to overload the + operator so that you can add two Rectangle objects. Adding two Rectangle objects should give another Rectangle object whose length is the sum of the lengths of the two Rectangle objects and whose breadth is the sum of the breadths of the two Rectangle objects.

Solution:

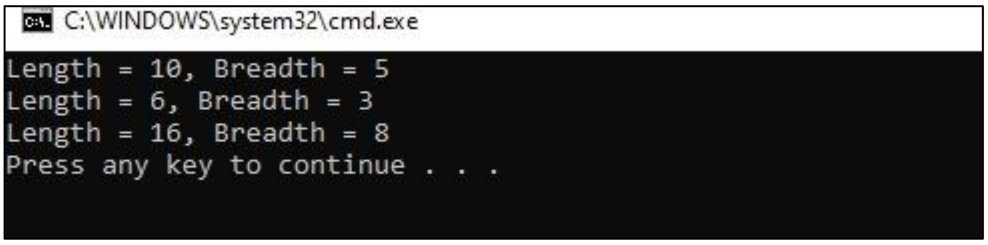
Main Method:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApp18
{
    class Program
    {
        static void Main(string[] args)
        {
            Rectangle r1 = new Rectangle(10, 5);
            Rectangle r2 = new Rectangle(6, 3);
            Rectangle r3 = r1 + r2;
            r1.Display();
            r2.Display();
            r3.Display();
        }
    }
    }
```

Class:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApp18
{
    class Rectangle
    {
        int length, breadth;
        public Rectangle(int length, int breadth)
        {
            this.length = length;
            this.breadth = breadth;

        }

        public Rectangle()
        {
            this.length = 0;
            this.breadth = 0;

        }

        public void Display()
        {
            Console.WriteLine("Length = " + this.length + ", Breadth = " + this.breadth);
        }

        public static Rectangle operator +(Rectangle r1, Rectangle r2)
        {
            Rectangle r3 = new Rectangle();
            r3.length = r1.length + r2.length;
            r3.breadth = r1.breadth + r2.breadth;
            return r3;
        }
    }
}
```

Output:



```
Length = 10, Breadth = 5
Length = 6, Breadth = 3
Length = 16, Breadth = 8
Press any key to continue . . .
```

Task No. 6: Write a class Time which represents time. the class should have three fields for hours, minutes and seconds. It should have constructor to initialize the hours, minutes and seconds.
A method printTime() to print the current time.
Overload the following operators:
plus operator (+) (add two time objects based on 24 hour clock)
and < (compare two time objects)

Solution:

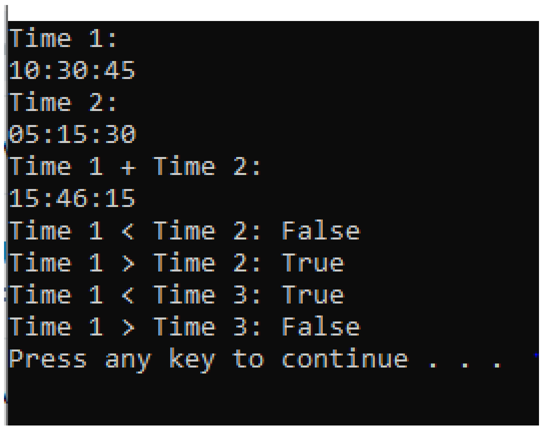
Main Method:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApp22
{
    class Program
    {
        static void Main(string[] args)
        {
            Time time1 = new Time(10, 30, 45);
            Time time2 = new Time(5, 15, 30);
            Time time3 = time1 + time2;

            Console.WriteLine("Time 1:");
            time1.PrintTime();

            Console.WriteLine("Time 2:");
            time2.PrintTime();

            Console.WriteLine("Time 1 + Time 2:");
            time3.PrintTime();

            Console.WriteLine("Time 1 < Time 2: " + (time1 < time2));
            Console.WriteLine("Time 1 > Time 2: " + (time1 > time2));
            Console.WriteLine("Time 1 < Time 3: " + (time1 < time3));
            Console.WriteLine("Time 1 > Time 3: " + (time1 > time3));
        }
    }
}
```

Class:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApp22
{
    class Time
    {
        int hours, minutes, seconds;
        public Time()
        {
            this.hours = 0;
            this.minutes = 0;
            this.seconds = 0;

        }
        public Time(int hours, int minutes, int seconds)
        {
            this.hours = hours;
            this.minutes = minutes;
            this.seconds = seconds;
        }
        public void PrintTime()
        {
            Console.WriteLine("{0:D2}:{1:D2}:{2:D2}", hours, minutes, seconds);
        }

        public static Time operator +(Time a, Time b)
        {
            Time T = new Time();
            T.hours = a.hours + b.hours;
            T.minutes = a.minutes + b.minutes;
            T.seconds = a.seconds + b.seconds;
            if (T.seconds >= 60)
            {
                T.minutes++;
                T.seconds -= 60;
            }
            if (T.minutes >= 60)
            {
                T.hours++;
                T.minutes -= 60;
            }
            if (T.hours >= 24)
            {

                T.hours -= 24;
            }
            return T;
        }
        public static bool operator <(Time a, Time b)
        {
            if (a.hours < b.hours)
            {
                return true;
            }
            else if (a.hours == b.hours && a.minutes < b.minutes)
            {
                return true;
            }
            else if (a.hours == b.hours && a.minutes == b.minutes && a.seconds < 
b.seconds)
            {
                return true;
            }
            else
            {
                return false;
            }

        }
        public static bool operator >(Time a, Time b)
        {
            if (a.hours > b.hours)
            {
                return true;
            }
            else if (a.hours == b.hours)
            {
                if (a.minutes > b.minutes)
                {
                    return true;
                }
                else if (a.minutes == b.minutes)
                {
                    if (a.seconds > b.seconds)
                    {
                        return true;
                    }
                    else
                    {
                        return false;
                    }
                }
                else
                {
                    return false;
                }
            }
            else
            {
                return false;
            }
        }
    }
}
```

<u>Output:</u>



```
Time 1:
10:30:45
Time 2:
05:15:30
Time 1 + Time 2:
15:46:15
Time 1 < Time 2: False
Time 1 > Time 2: True
Time 1 < Time 3: True
Time 1 > Time 3: False
Press any key to continue . . .
```