

Method overloading

Method overloading:- *It has same method name with different argument lists. It is use to same memory at compile time It is also known as compile time polymorphism. It creates user friendly environment.*

Example 1:-

```
public class Sum {  
  
    public int sum(int x, int y)  
    {  
        return (x + y);  
    }  
  
    public int sum(int x, int y, int z)  
    {  
        return (x + y + z);  
    }  
  
    public double sum(double x, double y)  
    {  
        return (x + y);  
    }  
  
    public static void main(String args[])  
    {  
        Sum X = new Sum();  
    }  
}
```

```
        System.out.println(X.sum(10, 20));  
        System.out.println(X.sum(10, 20, 20));  
        System.out.println(X.sum(11.0, 20.5));  
    }  
}
```

Output:-

30
50
31.5

Example 2:-

```
public class Sum {  
  
    public int sum(int x, int y, int z)  
    {  
        return (x + y);  
    }  
  
    public int sum(int x, int y)  
    {  
        return (x + y + z);  
    }  
  
    public double sum(double x, double y)  
    {  
        return (x + y);  
    }  
}
```

```
}  
  
public static void main(String args[])  
{  
    Sum X = new Sum();  
    System.out.println(X.sum(110, 20));  
    System.out.println(X.sum(10, 10, 10));  
    System.out.println(X.sum(11.0, 20.5));  
}  
}
```

Output:-

130

30

31.5

Created by:- Ajay kumar verma