Your task is to create a Java class Product that encapsulates the following details of a product:

- productID (int)
- productName (String)
- price (double)
- quantity (int)

The company needs the ability to:

Set and get the values for these attributes using appropriate getter and setter methods.
Ensure that the price cannot be negative or zero, and the quantity cannot be negative.

Requirements:

- Implement the class Product using proper encapsulation.
- Add logic to the setter methods to validate the price and quantity.
- Write a main method that creates a Product object, sets valid values, tries to set invalid values (to see if validation works), and displays the product details.

```
package samplepackage;

class Product
{
    private int productId;
    private String productName;
    private double price;
    private double qty;

    public int getProductId() {
        return productId;
    }

    public void setProductId(int productId) {
        this.productId = productId;
    }
    public String getProductName() {
        return productName;
    }

    public void setProductName(String productName) {
        this.productName = productName;
    }

    public double getPrice() {
        return price;
    }

    public void setPrice(double price) {

        if(price > 0)
        {
            this.price = price;
        }
        else
        {
            System.err.println("Value should be above 0");
        }

    }

    public double getQty() {
        return qty;
    }

    public void setQty(double qty) {
        if(qty >=0)
        {
           this.qty = qty;
        }
        else
        {
            System.err.println("Qty cannot be a negative value!");
        }

    }




}

public class DriverClass {

    public static void main(String[] args) {
        Product product1 = new Product();
        product1.setProductId(1);
        product1.setProductName("ABC");
        product1.setPrice(100);
        product1.setQty(2);

        System.out.println("Total - "+(product1.getPrice() * product1.getQty()));

    }

}
```

Constructor overloading

```
package samplepackage;

class Item
{

   private int itemNo;
   private double itemPrice;
   //Constructor overloading
   public Item() //Paramterized Constructor
   {
      itemNo = 0;
      itemPrice = 0;
   }
   public Item(int a,double b) //Paramterized Constructor
   {
      itemNo = a;
      itemPrice = b;
   }
   public Item(int a) //Paramterized Constructor
   {
      itemNo = a;
      itemPrice = 0;
   }
}

public class DriverClass {

    public static void main(String[] args) {
        Item obj = new Item(12);

    }

}
```