

Dada la siguiente clase PRODUCTO y la clase Fecha (que ya esta resuelta y de la cual no debe ocuparse)

```
class PRODUCTO
```

```
{
```

```
private:
```

```
char *    Nombre;
```

```
protected:
```

```
int       Codigo;
```

```
public:
```

```
/*constructor parametrizado con valores  
iniciales triviales*/
```

```
/*constructor de copia*/
```

```
/*destructor (inline)*/
```

```
void Set_Datos(char*, int);
```

```
void Set_Nombre (char*);
```

```
char* Get_Nombre(void);
```

```
/*sobrecarga del operador << */
```

```
};
```

```
class Fecha {
```

```
public:
```

```
Fecha( int , int , int );
```

```
Fecha( Fecha& );
```

```
friend ostream&  
operator<<(ostream&, Fecha& );
```

```
~Fecha();
```

```
private:
```

```
int mes;      // 1-12
```

```
int dia;      // 1-31
```

```
int anio;
```

```
};
```

Se pide:

1. Defina TODOS los métodos de la clase “PRODUCTO” .

Como aclaración: Debe hacer uso del concepto de memoria dinámica.

En caso de tener que inicializar o modificar un objeto de la clase sin contar con datos precisos, deberá ponerle el nombre “SIN NOMBRE”, y 0 al código.

La sobrecarga del operador << tiene que devolver la siguiente cadena: “PRODUCTO NOMBRE: XXXX\t\tCODIGO: XXXX\n”, siendo XXXX los valores correspondientes al objeto.

2. Declare una nueva clase llamada “PRODUCTOenSTOCK” que herede todos los miembros de la primera, adicionando tres miembros privados ***unsigned int Cantidad, char* Proveedor y Fecha fecha_compra***, que deberán amanecer en cero, “SIN PROVEEDOR” y en 1 de enero de 2000, al ser instanciada la clase sin parámetros. El constructor de esta clase debe ser definido inline.

Desarrolle un constructor parametrizado y otro de copia de la clase PRODUCTOenSTOCK

3. Genere un nuevo método **void Set_Datos** en la clase PRODUCTOenSTOCK, que almacene los valores recibidos en Cantidad, Proveedor y Fecha y que actualice también los miembros heredados de la clase base.
4. Agregue en la clase PRODUCTOenSTOCK otros dos métodos que sean operator++ y operator— (ambos posincremento), que sumen o resten la cantidad de los productos en el stock.