

```
1  /*
2   * Proyecto:  PlantillaDeClases
3   * Archivo:   Pantilla.h
4   * Autor:    J. Miguel Guanira E.
5   *
6   * Created on 12 de noviembre de 2024, 08:08 AM
7   */
8
9
10 #ifndef PANTILLA_H
11 #define PANTILLA_H
12 #include <iostream>
13 #include <iomanip>
14 using namespace std;
15
16 template <typename T>
17 class Pantilla {
18 private:
19     T dato;
20 public:
21     void SetDato(T dato);
22     T GetDato() const;
23     void leerDato();
24     void imprimeDato();
25 };
26
27 template <typename T>
28 void Pantilla<T>::SetDato(T d) {
29     dato = d;
30 }
31
32 template <typename T>
33 T Pantilla<T>::GetDato() const {
34     return dato;
35 }
36
37 template <typename T>
38 void Pantilla<T>::leerDato() {
39     cin>>dato;
40 }
41
42 template <typename T>
43 void Pantilla<T>::imprimeDato() {
44     cout<<dato<<endl;
45 }
46
47 #endif /* PANTILLA_H */
48
49 /*
50 * Proyecto:  PlantillaDeClases
51 * Archivo:   Persona.h
52 * Autor:    J. Miguel GuaniraErazo (Juan Miguel)
53 *
54 * Creado el 30 de junio de 2020, 06:23 PM
55 */
56
57 #ifndef PERSONA_H
58 #define PERSONA_H
59 #include <iostream>
60 class Persona {
61 private:
62     int dni;
63     char*nombre;
64     double sueldo;
65 public:
66     Persona();
```

```
67     Persona(const Persona& orig);
68     virtual ~Persona();
69     void SetSueldo(double sueldo);
70     double GetSueldo() const;
71     void SetNombre(const char* nombre);
72     void GetNombre(char* ) const;
73     void SetDni(int dni);
74     int GetDni() const;
75     void operator =(const class Persona&);
76 };
77 void operator >> (istream &, class Persona&);
78 ostream& operator << (ostream &,const class Persona&);
79 #endif /* PERSONA_H */
80
81 /*
82  * Proyecto:  PlantillaDeClases
83  * Archivo:  Persona.cpp
84  * Autor:    J. Miguel GuaniraErazo (Juan Miguel)
85  *
86  * Creado el 30 de junio de 2020, 06:23 PM
87  */
88 #include <iostream>
89 #include <iomanip>
90 using namespace std;
91 #include <cstring>
92 #include "Persona.h"
93
94 Persona::Persona() {
95     nombre = nullptr;
96 }
97
98 Persona::Persona(const Persona& orig) {
99     nombre = nullptr;
100     *this = orig;
101 }
102
103 Persona::~~Persona() {
104     if (nombre != nullptr) delete nombre;
105 }
106
107 void Persona::SetSueldo(double sueldo) {
108     this->sueldo = sueldo;
109 }
110
111 double Persona::GetSueldo() const {
112     return sueldo;
113 }
114
115 void Persona::SetNombre(const char* nomb) {
116     if (nombre != nullptr) delete nombre;
117     nombre = new char[strlen(nomb)+1];
118     strcpy(nombre,nomb);
119 }
120
121 void Persona::GetNombre(char*nomb) const {
122     if(nombre == nullptr) nomb[0]=0;
123     else strcpy(nomb,nombre);
124 }
125
126 void Persona::SetDni(int dni) {
127     this->dni = dni;
128 }
129
130 int Persona::GetDni() const {
131     return dni;
132 }
```

```
133
134 void Persona::operator =(const class Persona&per) {
135     char nomb[60];
136     dni = per.dni;
137     per.GetNombre(nomb);
138     SetNombre(nomb);
139     sueldo = per.sueldo;
140 }
141
142 void operator >> (istream &in, class Persona&per) {
143     int dni;
144     char nomb[60];
145     double sueldo;
146     in>>dni;
147     in.get(); //coma
148     in.getline(nomb,60,',');
149     in>>sueldo;
150     per.SetDni(dni);
151     per.SetNombre(nomb);
152     per.SetSueldo(sueldo);
153 }
154
155 ostream& operator << (ostream &out,const class Persona&per) {
156     out.precision(2);
157     out<<fixed;
158     char nomb[60];
159     per.GetNombre(nomb);
160     out<<left<<setw(10)<<per.GetDni()<<setw(30)<<nomb
161         <<right<<setw(10)<<per.GetSueldo();
162     return out;
163 }
164
165 /*
166  * Proyecto: PlantillaDeClases
167  * Archivo:  main.cpp
168  * Autor:    J. Miguel Guanira E.
169  *
170  * Created on 11 de noviembre de 2024, 09:22 PM
171  */
172
173 #include <iostream>
174 #include <iomanip>
175 #include <string>
176 using namespace std;
177 #include "Pantilla.h"
178 #include "Persona.h"
179
180 int main(int argc, char** argv) {
181     class Pantilla<int>plantillaInt;
182     plantillaInt.leerDato();
183     plantillaInt.imprimeDato();
184
185     // string cadena;
186     class Pantilla<string> plantillaStr;
187     plantillaStr.leerDato();
188     plantillaStr.imprimeDato();
189
190     class Pantilla<class Persona> plantillaPer;
191     plantillaPer.leerDato();
192     plantillaPer.imprimeDato();
193
194     return 0;
195 }
196
197
```