

```

1: ///////////////////////////////////////////////////////////////////
2: //
3: // Fundamentos de Programación
4: // ETS Informática y Telecomunicaciones
5: // Universidad de Granada
6: // // Don Oreó
7: //
8: ///////////////////////////////////////////////////////////////////
9:
10: //Cuadrado con constructor
11:
12: #include <iostream>
13: #include <cmath>
14: using namespace std;
15:
16: class Cuadrado{
17: private:
18:     double x,y;
19:     double lado;
20: public:
21:     Cuadrado(double coord_x, double coord_y, double longitud){
22:         if(longitud > 0){
23:             x = coord_x;
24:             y = coord_y;
25:             lado = longitud;
26:         }
27:         else{
28:             x = NAN;
29:             y = NAN;
30:             lado = -1;
31:         }
32:     }
33:
34:     void SetCoordenadasLado(double coord_x, double coord_y, double longitud){
35:         x = coord_x;
36:         y = coord_y;
37:         lado = longitud;
38:     }
39:
40:     double Coord_x(){
41:         return x;
42:     }
43:
44:     double Coord_y(){
45:         return y;
46:     }
47:
48:     double Lado(){
49:         return lado;
50:     }
51:
52:     double Area(){
53:         return lado*lado;
54:     }
55:
56:     double Perimetro(){
57:         return 4*lado;
58:     }
59:
60: };
61:
62: int main(){
63:
64:     const string MSJ_COORDENADAS = "\nCoordenadas: ";
65:     const string MSJ_LONGITUD = "\nLongitud: ";
66:     const string MSJ_AREA = "\nÁrea: ";
67:     const string MSJ_PERIMETRO = "\nPerímetro: ";
68:     const string MSJ_PARCELA = "\nParcela ";
69:     double esquina_x1, esquina_y1, lado1,
70:         esquina_x2, esquina_y2, lado2;
71:
72:     cout << "Introduce las coordenadas de la esquina inferior izquierda del cuadrado 1 y su lado: " << endl;
73:     cin >> esquina_x1
74:         >> esquina_y1
75:         >> lado1;
76:
77:     Cuadrado una_parcela(esquina_x1, esquina_y1, lado1);
78:
79:     cout << "Introduce las coordenadas de la esquina inferior izquierda del cuadrado 2 y su lado: " << endl;
80:     cin >> esquina_x2
81:         >> esquina_y2
82:         >> lado2;
83:
84:     Cuadrado otra_parcela(esquina_x2, esquina_y2, lado2);

```

```
85:
86:     cout << MSJ_PARCELA << "1";
87:     cout << MSJ_COORDENADAS << una_parcela.Coord_x() << " , " << una_parcela.Coord_y();
88:     cout << MSJ_LONGITUD << una_parcela.Lado();
89:     cout << MSJ_AREA << una_parcela.Area();
90:     cout << MSJ_PERIMETRO << una_parcela.Perimetro() << endl;
91:
92:     cout << MSJ_PARCELA << "2";
93:     cout << MSJ_COORDENADAS << otra_parcela.Coord_x() << " , " << otra_parcela.Coord_y();
94:     cout << MSJ_LONGITUD << otra_parcela.Lado();
95:     cout << MSJ_AREA << otra_parcela.Area();
96:     cout << MSJ_PERIMETRO << otra_parcela.Perimetro() << endl;
97:
98:
99:     return 0;
100: }
101: /*
102: 3.4 5.7 2.9
103: -5.6 -4.1 1.8
104: */
105: /*
106: Coordenadas: 3.4 , 5.7
107: Longitud: 2.9
108: Área: 8.41
109: Perímetro: 11.6
110:
111: Coordenadas: -5.6 -4.1
112: Longitud: 1.8
113: Área: 3.24
114: Perímetro: 7.2
115: */
116:
```

```
1: ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
2: //
3: // Fundamentos de Programación
4: // ETS Informática y Telecomunicaciones
5: // Universidad de Granada
6: // // Don Oreó
7: //
8: ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
9:
10: //Cuadrado
11:
12: #include <iostream>
13: using namespace std;
14:
15: class Cuadrado{
16: private:
17:     double x,y;
18:     double lado;
19: public:
20:     void SetCoordenadasLado(double coord_x, double coord_y, double longitud){
21:         x = coord_x;
22:         y = coord_y;
23:         lado = longitud;
24:     }
25:
26:     double Coord_x(){
27:         return x;
28:     }
29:
30:     double Coord_y(){
31:         return y;
32:     }
33:
34:     double Lado(){
35:         return lado;
36:     }
37:
38:     double Area(){
39:         return lado*lado;
40:     }
41:
42:     double Perimetro(){
43:         return 4*lado;
44:     }
45:
46: };
47:
48: int main(){
49:     Cuadrado una_parcela;
50:     Cuadrado otra_parcela;
51:
52:     double esquina_x1, esquina_y1, lado1,
53:         esquina_x2, esquina_y2, lado2;
54:
55:     cout << "Introduce las coordenadas de la esquina inferior izquierda del cuadrado 1 y su lado: " << endl;
56:     cin >> esquina_x1 >> esquina_y1 >> lado1;
57:
58:     una_parcela.SetCoordenadasLado(esquina_x1, esquina_y1, lado1);
59:
60:     cout << "Introduce las coordenadas de la esquina inferior izquierda del cuadrado 2 y su lado: " << endl;
61:     cin >> esquina_x2 >> esquina_y2 >> lado2;
62:
63:     otra_parcela.SetCoordenadasLado(esquina_x2, esquina_y2, lado2);
64:
65:     cout << "Coordenadas: " << una_parcela.Coord_x() << " , " << una_parcela.Coord_y() << endl;
66:     cout << "Longitud: " << una_parcela.Lado() << endl;
67:     cout << "Area: " << una_parcela.Area() << endl;
68:     cout << "Perimetro " << una_parcela.Perimetro() << endl;
69:     cout << "Coordenadas: " << otra_parcela.Coord_x() << " , " << otra_parcela.Coord_y() << endl;
70:     cout << "Longitud: " << otra_parcela.Lado() << endl;
71:     cout << "Area: " << otra_parcela.Area() << endl;
72:     cout << "Perimetro: " << otra_parcela.Perimetro() << endl;
73:
74:
75:
76: }
```

```

1: ///////////////////////////////////////////////////////////////////
2: //
3: // Fundamentos de Programación
4: // ETS Informática y Telecomunicaciones
5: // Universidad de Granada
6: // // Don Oreó
7: //
8: ///////////////////////////////////////////////////////////////////
9:
10: //Dinero Datos miembro Privados
11:
12: #include <iostream>
13: using namespace std;
14:
15: class Dinero{
16: private:
17:     int euros;
18:     int centimos;
19:
20: public:
21:     void SetEurCent (int eurs,int cents){
22:         while (cents>=100){
23:             eurs++;
24:             cents-=100;
25:         }
26:         euros = eurs;
27:         centimos = cents;
28:     }
29:
30:     int Euros() {
31:         return euros;
32:     }
33:     int Centimos() {
34:         return centimos;
35:     }
36:
37: } ;
38:
39: int main() {
40:     Dinero un_dinero;
41:     Dinero otro_dinero;
42:     Dinero dinero_total;
43:
44:     int euros1,centimos1,
45:         euros2,centimos2;
46:     int eursuma,centsuma;
47:     cout << "Introduce el primer dinero: " << endl;
48:     cin >> euros1 >> centimos1;
49:     cout << "Introduce el segundo dinero: " << endl;
50:     cin >> euros2 >> centimos2;
51:
52:     un_dinero.SetEurCent (euros1,centimos1);
53:     otro_dinero.SetEurCent (euros2,centimos2);
54:
55:     euros1 = un_dinero.Euros();
56:     centimos1 = un_dinero.Centimos();
57:     euros2 = otro_dinero.Euros();
58:     centimos2 = otro_dinero.Centimos();
59:
60:     eursuma = euros1 + euros2;
61:     centsuma = centimos1 + centimos2;
62:
63:     dinero_total.SetEurCent (eursuma,centsuma);
64:
65:     cout << "En el primer dinero hay " << un_dinero.Euros() << "," << un_dinero.Centimos() << " euros" <<
endl;
66:     cout << "En el segundo dinero hay " << otro_dinero.Euros() << "," << otro_dinero.Centimos() << " euro
s" << endl;
67:     cout << "La suma del dinero es: " << dinero_total.Euros() << "," << dinero_total.Centimos() << " euro
s" << endl;
68:
69: }

```

```

1: ///////////////////////////////////////////////////////////////////
2: //
3: // Fundamentos de Programación
4: // ETS Informática y Telecomunicaciones
5: // Universidad de Granada
6: // // Don Oreó
7: //
8: ///////////////////////////////////////////////////////////////////
9:
10: //Dinero Datos miembro Publicos
11:
12: #include <iostream>
13: using namespace std;
14:
15: class Dinero{
16: public:
17:     int euros;
18:     int centimos;
19:
20:     void SetEurCent (int eurs,int cents){
21:         while (cents>=100){
22:             eurs++;
23:             cents-=100;
24:         }
25:         euros = eurs;
26:         centimos = cents;
27:     }
28:
29: } ;
30:
31: int main() {
32:     Dinero un_dinero;
33:     Dinero otro_dinero;
34:     Dinero dinero_total;
35:
36:     int euros1,centimos1,
37:         euros2,centimos2;
38:     int eursuma,centsuma;
39:     cout << "Introduce el primer dinero: " << endl;
40:     cin >> euros1 >> centimos1;
41:     cout << "Introduce el segundo dinero: " << endl;
42:     cin >> euros2 >> centimos2;
43:
44:     un_dinero.SetEurCent (euros1,centimos1);
45:     otro_dinero.SetEurCent (euros2,centimos2);
46:
47:     eursuma = un_dinero.euros + otro_dinero.euros;
48:     centsuma = un_dinero.centimos + otro_dinero.centimos;
49:
50:     dinero_total.SetEurCent (eursuma,centsuma);
51:
52:     cout << "En el primer dinero hay " << un_dinero.euros << "," << un_dinero.centimos << " euros" << endl;
53:     cout << "En el segundo dinero hay " << otro_dinero.euros << "," << otro_dinero.centimos << " euros" << endl;
54:     cout << "La suma del dinero es: " << dinero_total.euros << "," << dinero_total.centimos << " euros" << endl;
55:
56: }

```

```

1: //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
2: //
3: // Fundamentos de Programación
4: // ETS Informática y Telecomunicaciones
5: // Universidad de Granada
6: // // Don Oreó
7: //
8: //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
9:
10: //FormateadoDoubles
11:
12: #include<iostream>
13: using namespace std;
14:
15: double Redondea(double real, int num_decimales){
16: ...
17: }
18:
19: string EliminaUltimos(string cadena, char ultimo_car){
20: ...
21: }
22:
23: class FormateadorDoubles(){
24: private:
25:     string izda = "";
26:     string dcha = "";
27:     int decimales = 2;
28:     enum class SeparadorDecimal {PUNTO, COMA};
29:     SeparadorDecimal separador = SeparadorDecimal::COMA;
30: public:
31:     FormateadorDoubles()
32:     {
33:     }
34:     FormateadorDoubles(string delim_izda, delim_dcha)
35:     {
36:     }
37:     void Izda(){
38:         return izda;
39:     }
40:
41:     void Dcha(){
42:         return dcha;
43:     }
44:     int SetDecimales(int num_decimales){
45:         decimales = num_decimales;
46:     }
47:     void SetSeparadorPunto(){
48:         separador = SeparadorDecimal::PUNTO;
49:     }
50:     string GetCadena(double real){
51:         ... Redondea, EliminaUltimos, etc
52:     }
53:
54: };
55:
56: int main(){
57:     double num;
58:
59:     cout << "Introduce un numero: ";
60:     cin >> num;
61:     format_real(); //o format_real("<", ">")
62:
63:     format_real.GetCadena();
64:
65: }
66:

```

```

1: //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
2: //
3: // Fundamentos de Programación
4: // ETS Informática y Telecomunicaciones
5: // Universidad de Granada
6: // // Don Oreó
7: //
8: //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
9:
10: //Interfaz Instante
11:
12: #include<iostream>
13: using namespace std;
14:
15: class Instante{
16: private:
17:     int s = 0;
18:     int m = 0;
19:     int h = 0;
20: public:
21:     //opcion 1 introduciendo los segundos
22:     Instante(int seg){
23:         if(seg >=0){
24:             s=seg;
25:         }
26:         else
27:             s=-1;
28:     }
29:     //Opcion 2 introduciendo horas minutos segundos
30:     Instante(int hor, int min, int seg){
31:         if(seg >= 0 && min >= 0 && hor >= 0){
32:             s = seg;
33:             m = min;
34:             h = hor;
35:         }
36:         else{
37:             s = -1;
38:             m = -1;
39:             h = -1;
40:         }
41:     }
42: }
43:
44: void SetHorMinSeg(){
45:     while(s >= 60){
46:         m++;
47:         s-=60;
48:         while(m >= 60){
49:             h++;
50:             m-=60;
51:             while(h >= 24)
52:                 h -= 24;
53:         }
54:     }
55: }
56:
57: int S(){
58:     return s;
59: }
60:
61: int M(){
62:     return m;
63: }
64:
65: int H(){
66:     return h;
67: }
68:
69: int SegTotales(){
70:     int s_total = 0;
71:     s_total = s + m*60 + h*3600;
72:
73:     return s_total;
74: }
75:
76: int Minutos(){
77:     int m_total = 0;
78:     m_total = m + h*60;
79:
80:     return m_total;
81: }
82:
83: void conversion_segundos(){
84:     while(s >= 60){
85:         m++;
86:         s-=60;

```

```
87:         while(m >= 60) {
88:             h++;
89:             m-=60;
90:             while(h >= 24)
91:                 h = 0;
92:         }
93:     }
94: }
95: }
96: };
97:
98: int main() {
99:
100:
101:     int s,m,h;
102:     cout << "Introduce los segundos: ";
103:     cin >> s;
104:     Instante tiempo0(s);
105:     tiempo0.conversion_segundos();
106:     cout << "Tiempo 0: " << tiempo0.H() << "h " << tiempo0.M() << "min " << tiempo0.S() << "seg" << endl;
107:
108:     cout << "Introduzca la primera hora en formato(s m h): " << endl;
109:     cin >> h >> m >> s;
110:     Instante tiempo1(h,m,s);
111:     tiempo1.SetHorMinSeg();
112:
113:     cout << "Introduzca la segunda hora en formato(s m h): " << endl;
114:     cin >> h >> m >> s;
115:     Instante tiempo2(h,m,s);
116:     tiempo2.SetHorMinSeg();
117:
118:     cout << "\n\n";
119:     cout << "Tiempo 1: " << tiempo1.H() << "h " << tiempo1.M() << "min " << tiempo1.S() << "seg" << endl;
120:     cout << "Tiempo 2: " << tiempo2.H() << "h " << tiempo2.M() << "min " << tiempo2.S() << "seg" << endl;
121:
122:     cout << "Segundos(t1): " << tiempo1.SegTotales() << " segundos."
123:         << "\nMinutos(t1): " << tiempo1.Minutos() << " minutos" << endl;
124:
125:     return 0;
126: }
127:
```



```

1: //////////////////////////////////////
2: //
3: // Fundamentos de Programación
4: // ETS Informática y Telecomunicaciones
5: // Universidad de Granada
6: // // Don Oreo
7: //
8: //////////////////////////////////////
9:
10: //Interfaz Secuencia Caracteres
11: /*
12: Se quiere definir una clase SecuenciaCaracteres similar a la clase string para
13: manipular secuencias de caracteres (de datos de tipo char).
14: Esto es una secuencia de caracteres
15: Como dato miembro privado, se recomienda usar un vector de caracteres. Ya sabemos
16: que debemos reservar memoria suficiente. Supondremos que el máximo será 100. En
17: vez de usar el literal 100, mejor usamos una constante. En ese caso, C++ obliga a que
18: sea una constante estática (si aún no ha visto las constantes estáticas en clase de
19: Teoría, no se preocupe ya que no se le pide que el programa compile). Nos quedaría
20: lo siguiente:
21:
22: private:
23:     static const int TAM = 100;
24:     char caracteres[TAM];
25: Se proponen dos alternativas para marcar el bloque del vector que se va a usar:
26: \225 Usar un terminador fijo, por ejemplo, #, al final del bloque usado. En el ejemplo
27: anterior, el vector caracteres contendría:
28: 'E' 's' 't' 'o' ' ' 'e' 's' .... 'r' 'e' 's' '#' ? ? .... ?
29: \225 Usar un dato miembro privado utilizados que nos diga cuántos elementos
30: estamos usando en cada momento. En el ejemplo anterior, utilizados valdría
31: 35.
32: ¿Qué opción le parece más adecuada? ¿Por qué?
33: El mejor es el segundo ya que el util es más manejable que el TERMINADOR
34: ¿Qué métodos definiría para manipular la secuencia? Al menos debe definir las cabeceras de los métodos p
ara realizar lo siguiente:
35: a) Método Aniade para añadir un carácter al final de la secuencia.
36: Tenga en cuenta que a una variable cadena de tipo string se le puede asignar
37: directamente cadena = "Hola". Sin embargo, eso no lo sabemos hacer por
38: ahora con objetos de nuestras propias clases. Por lo tanto, la única forma de
39: añadir caracteres a un objeto de la clase SecuenciaCaracteres sería hacerlo
40: de uno en uno. Así pues, llamando a un método Aniade, añadiríamos la 'H',
41: luego la 'o' y así sucesivamente.
42: b) Método Utilizados para obtener la longitud actual de la secuencia (el número
43: de caracteres que contiene)
44: c) Método Invierte para invertir la secuencia. En el ejemplo anterior, la secuencia
45: se quedaría en:
46: seretcarac ed aicneuces anu se otsE
47: d) Método PrimeraOcurrencia para buscar la primera ocurrencia de un carácter.
48: e) Método EliminaOcurrencias para eliminar las ocurrencias de un carácter.
49: Por ejemplo, después de eliminar el carácter 'a', la secuencia quedaría así:
50: Esto es un secuenci de crcteres
51: */
52: #include <iostream>
53: using namespace std;
54:
55: class SecuenciaCaracteres{
56: private:
57:     static const int TAMANIO = 100;
58:     char caracteres[TAMANIO];
59:     int utilizados = 0;
60: public:
61:     void Aniade(char car){
62:         caracteres[utilizados] = car;
63:         utilizados++;
64:     }
65:
66:     int Utilizados(){
67:         return utilizados;
68:     }
69:
70:     char Elemento(int posicion){
71:         return caracteres[posicion];
72:     }
73:
74:     void Invierte(){
75:         char secuencia[TAMANIO];
76:         int pos;
77:         for(int i = 0; i <= utilizados; i++){
78:             secuencia[i] = caracteres[i];
79:
80:             pos = 0;
81:             for(int j = utilizados-1; j >= 0; j--){
82:                 caracteres[pos] = secuencia[j];
83:                 pos++;
84:             }
85:             cout << "\n";

```

```

86:     for(int i = 0; i < utilizados; i++)
87:         cout << caracteres[i];
88:     }
89:
90:     int PrimeraOcurrecencia(char car){
91:         bool encontrado = 0;
92:         int pos_ocurrencia = -1;
93:         for(int i = 0; i < utilizados && ! encontrado; i++){
94:             if(caracteres[i] == car){
95:                 pos_ocurrencia = i+1;
96:                 encontrado = true;
97:             }
98:         }
99:         return pos_ocurrencia;
100:     }
101:     void EliminaOcurrencias(char a_borrar){
102:         char secuencia[TAMANIO];
103:         int pos;
104:
105:         pos = 0;
106:         for(int i = 0; i < utilizados; i++){
107:             if(caracteres[i] != a_borrar ){
108:                 secuencia[pos] = caracteres[i];
109:                 pos++;
110:             }
111:         }
112:         utilizados = pos;
113:         cout << "\n";
114:         for(int i = 0; i < utilizados; i++){
115:             caracteres[i] = secuencia[i];
116:             cout << caracteres[i];
117:         }
118:     }
119:
120:     void ImprimeSecuencia(){
121:         for(int i=0; i<utilizados; i++)
122:             cout << caracteres[i];
123:     }
124:
125: };
126:
127: int main(){
128:
129:     SecuenciaCaracteres Prueba;
130:
131:     Prueba.Aniade('H');
132:     Prueba.Aniade('o');
133:     Prueba.Aniade('l');
134:     Prueba.Aniade('a');
135:
136:     Prueba.ImprimeSecuencia();
137:
138:     Prueba.EliminaOcurrencias('H');
139:
140:     return 0;
141: }

```

```

1: //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
2: //
3: // Fundamentos de Programación
4: // ETS Informática y Telecomunicaciones
5: // Universidad de Granada
6: // // Don Oreó
7: //
8: //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
9:
10: //Interfaz Simulador Depósito
11:
12: #include<iostream>
13: #include<cmath>
14: using namespace std;
15:
16: class SimuladorDeposito{
17: private:
18:     double capital;
19:     double interes;
20: public:
21:     SimuladorDeposito()
22:     :capital(0),interes(0)
23:     {
24:     }
25:     void SetCapital(double capital_inicial){
26:         if(esPositivo)
27:             ....
28:     }
29:     void SetInteres(double I){
30:         if(esPositivo)
31:             ....
32:     }
33:     double Capital(){
34:         return capital;
35:     }
36:     int Interes(){
37:         return interes;
38:     }
39:     void CapitalFinal(int tope_anio){
40:         ....
41:     }
42:
43:     void DoblarCapital(){
44:         ....
45:     }
46: }
47:
48: };
49:
50: int main(){
51:     SimuladorDeposito sueldo1;
52:
53:     double capital_inicial;
54:     int interes;
55:     int anios_a_invertir;
56:
57:     cout << "Introduce el capital y el interes: ";
58:     cin >> capital_inicial >> interes;
59:
60:     sueldo1.SetCapital(capital_inicial);
61:     sueldo1.SetInteres(interes);
62:
63:     cout << "Introduce los años a invertir: ";
64:     cin >> anios_a_invertir;
65:
66:     sueldo1.CapitalFinal(anios_a_invertir);
67:     sueldo1.DoblarCapital();
68: }

```

```

1: ///////////////////////////////////////////////////////////////////
2: // Fundamentos de Programación
3: // ETS Informática y Telecomunicaciones
4: // Universidad de Granada
5: // Departamento de Ciencias de la Computación e Inteligencia Artificial
6: // Autor: Juan Carlos Cubero
7: ///////////////////////////////////////////////////////////////////
8:
9: // Generador aleatorio de números enteros
10:
11:
12: #include <random> // para la generación de números pseudoaleatorios
13: #include <chrono> // para la semilla
14: #include <iostream>
15: using namespace std;
16:
17:
18: class GeneradorAleatorioEnteros{
19: private:
20:     mt19937 generador_mersenne; // Mersenne twister
21:     uniform_int_distribution<int> distribucion_uniforme;
22:
23:     long long Nanosec(){
24:         return chrono::high_resolution_clock::now().time_since_epoch().count();
25:     }
26:
27: public:
28:     GeneradorAleatorioEnteros()
29:         :GeneradorAleatorioEnteros(0, 1){
30:     }
31:
32:     GeneradorAleatorioEnteros(int min, int max){
33:         const int A_DESCARTAR = 70000; // Panneton et al. ACM TOMS Volume 32 Issue 1, March 2006
34:         auto semilla = Nanosec();
35:
36:         generador_mersenne.seed(semilla);
37:         generador_mersenne.discard(A_DESCARTAR);
38:         distribucion_uniforme = uniform_int_distribution<int> (min, max);
39:     }
40:
41:     int Siguiente(){
42:         return distribucion_uniforme(generador_mersenne);
43:     }
44: };
45:
46:
47: int main(){
48:     int num_a_generar;
49:     GeneradorAleatorioEnteros aleat_num_a_generar(1,5);
50:     GeneradorAleatorioEnteros aleat_0_1;
51:
52:     for (int i = 0; i < 4; i++){
53:         num_a_generar = aleat_num_a_generar.Siguiente();
54:         cout << "\n El numero aleatorio entre 1 y 5 es: " << num_a_generar << endl;
55:         for(int j = 0; j < num_a_generar; j++){
56:             cout << " " << aleat_0_1.Siguiente();
57:         }
58:     }
59:
60: }

```

```
1: //////////////////////////////////////
2: //
3: // Fundamentos de Programación
4: // ETS Informática y Telecomunicaciones
5: // Universidad de Granada
6: // // Don Oreó
7: //
8: //////////////////////////////////////
9:
10: //Simulador Depósito
11:
12: #include<iostream>
13: #include<cmath>
14: using namespace std;
15:
16: class SimuladorDeposito{
17: private:
18:     double capital;
19:     double interes;
20: public:
21:     SimuladorDeposito()
22:     :capital(0),interes(0)
23:     {
24:     }
25:     void SetCapital(double capital_inicial){
26:         if(capital_inicial >= 0)
27:             capital = capital_inicial;
28:     }
29:     void SetInteres(double I){
30:         if(I >= 0)
31:             interes = I;
32:     }
33:     double Capital(){
34:         return capital;
35:     }
36:     int Interes(){
37:         return interes;
38:     }
39:     void CapitalFinal(int tope_anio){
40:         double elev;
41:         double int_compuesto;
42:         for(int anios = 0; anios < tope_anio;anios++){
43:             elev = pow((1 + interes/100),anios+1);
44:             int_compuesto = capital*elev;
45:             cout << "Capital obtenido transcurrido el año número " << anios
46:                 << " = " << int_compuesto << endl;
47:         }
48:     }
49: }
50:
51: void DoblarCapital(){
52:     double tope_cap = 2*capital;
53:     int anios;
54:
55:     while(capital<=tope_cap){
56:         capital = capital + capital*(interes/100);
57:         anios++;
58:     }
59:
60:     cout << "\nPara doblar la cantidad inicial han de pasar " << anios << " años" << endl;
61:     cout << "Al finalizar, se obtendrá un total de " << capital << " euros" << endl;
62: }
63:
64: };
65:
66:
67: int main(){
68:     SimuladorDeposito sueldo1;
69:
70:     double capital_inicial;
71:     int interes;
72:     int anios_a_invertir;
73:
74:     cout << "Introduce el capital y el interes: ";
75:     cin >> capital_inicial >> interes;
76:
77:     sueldo1.SetCapital(capital_inicial);
78:     sueldo1.SetInteres(interres);
79:
80:     cout << "Introduce los años a invertir: ";
81:     cin >> anios_a_invertir;
82:
83:     sueldo1.CapitalFinal(anios_a_invertir);
84:     sueldo1.DoblarCapital();
85: }
```