

<b>INFORMATICA II</b>	SUBTEMA	TEMA #	PARCIA L	<b><u>18/Julio/2020</u></b>
	C++	<b>5</b>	<b># 1</b>	

Estimado alumno,

A partir de la siguiente página encontrará el enunciado de su examen. Le recordamos que ya no puede cambiarlo.

Ud. deberá resolver el ejercicio en su computadora. Luego, deberá subir los archivos realizados comprimidos (preferentemente zip). Si lo desea, puede agregar un documento de texto con explicaciones que quiera hacerle llegar a su profesor.

Cada aproximadamente una hora debe actualizar su repositorio GIT con lo que tenga hasta ese momento resuelto.

Le recordamos:

	Fecha	tópico	Examen disponible durante	Tiempo para hacerlo desde bajada	Modalidad	Entregables
1	Sábado 18 de julio – 14hs.	C++	24hs.	4hs. (*)	ejercicio en compilador	Archivo .zip (ver NOTA 1)
2	Miércoles 22 de julio – 14hs.	IPCs	24hs.	2hs. (*)	cuestionario Moodle + ejercicio	<ul style="list-style-type: none"> <li>• Responder cuestionario Moodle</li> <li>• Archivo .zip (ver NOTA 1)</li> </ul>
3	Sábado 25 de julio – 14hs.	MdE	24hs.	4hs. (*)	ejercicio en uModel Factory	Archivo .zip (ver NOTA 2)

Cátedra de Info2

Nombre y Apellido	Nº Legajo	Calificación

Un taller mecánico automotor , utiliza un SG(sist Gestión) para el control de ingreso de los vehículos y la facturación de los trabajos realizados. Para ello utiliza las siguientes clases (la clase String ya está definida de la misma manera que std::string, solo úsela)

```
class Taller
{
String Marca;
String Modelo;
String Patente;
Hora    h;

public:
    Taller(); //Inicializa a las variables Marca.Modelo y Patente con la leyenda SIN
              DATOS y la Hora en 0,0
    Taller(const String&, const String&, const String&, int, int ); // inicializa a
                          Marca.Modelo y Patente y la hora de entrada
    // agregar mas métodos si considera que hace falta
    xxxxx String Tarea[20]; //tareas PREDEFINIDAS que deben ser cargadas ej
"C_Bujias", "Afinación", "Embrague", "C_Amortiguadores"
    xxxxx int Costo[20]; // costo de las tareas indicadas Por ej 2000,5000,8800,3500
    friend ostream& operator<< (ostream&, Taller&);
};
```

Donde xx atributos que ud debe declarar

```
class Hora
{
    int hora,min;
    public:
        Hora(); // inicializa en 0,0
        Hora(int,int=0);
    // agregar mas métodos si considera que hace falta
}
```

```
class Reparacion1 : public Taller
{
    struct Arreglo {
        int Indice; // indice de la tabla
        int Cant;   // cantidad de repuestos de la tarea
    } *Trabajos;
    int cant_arreglos; //Cantidad total de arreglos en el vehículo
    int Costo_tot; //costo total= Costo del ítem * Cant de repuestos
    public:
        Reparacion1();
        Reparacion1(const String&, const String&, const String&, int, int, int, int);
        //marca, modelo, patente, hora, min, indice, cantidad de rep
        friend ostream& operator<< (ostream &, Reparacion1& );
        void Tarea(int,int); // se carga el indice cada tarea y la cant de repuestos
                          // y muestra en pantalla la tarea y los costos parciales y
                          totales
        // agregar más métodos si considera que hacen falta
};
```

La clase Taller se encuentra diseñada y desarrollada de acuerdo a lo que se muestra más arriba. La misma hace referencia a un vehículo que entra al taller, indicando su marca, modelo, y patente. A su vez, para cualquier vehículo que entra al taller hay una serie de tareas básicas cuyos nombres se encuentran cargados en el vector de strings

**Tarea**, y cuyo costo unitario se encuentra indicado en el vector **Costo** (el mismo se multiplicará luego por la cantidad de partes que se necesiten para la reparación. Por ejemplo, si el costo de “Afinación” es de \$3000 y se necesitan 2 partes, el costo total de esta tarea será de \$6000)

El jefe de taller, utiliza la clase **Reparacion** para dar de alta un nuevo trabajo en el SG: Indica la marca, modelo y patente del auto, e indica, para cada Tarea, cuantas partes se necesitarán para su arreglo.

Teniendo en cuenta que muchas reparaciones consisten de una sola tarea, se permite cargar dicha tarea en el constructor parametrizado de la clase.

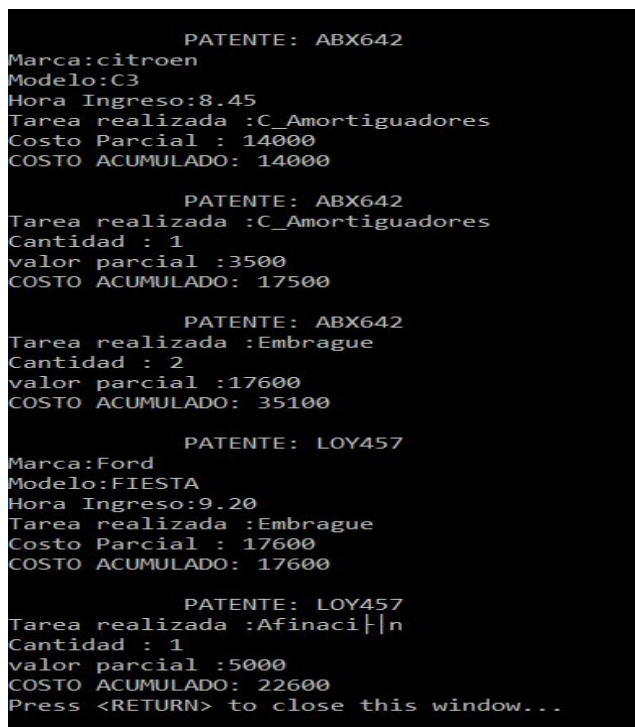
El método Tarea permite cargar al objeto instanciado, los diferentes trabajos de a uno a la vez en el vector dinámico **Trabajos** (indicando su índice, en relación al vector de **Tarea** y de **Costo**) y la cantidad de repuestos necesarios para cada trabajo. La variable **Costo\_tot** debe acumular el costo de las tareas realizadas en el vehículo

Se solicita que ud resuelva la clase Reparacion, cual deriva de la clase **Taller**, **utilizando los recursos de la misma**, sin generar redundancias con métodos propios agregando los atributos y métodos necesarios tal de poder cumplir con lo solicitado en el main()

```
int main(void)
{
    Reparacion1 M1("citroen","C3","ABX642",8,45,3,4); //marca-modelo-patente-hora
                                                    //indice de la tarea ,cant_rep
    cout<<M1;
    M1.Tarea(3,1); // Se carga la tarea de índice 3, con la cantidad de repuestos 1
    M1.Tarea(2,2); // Se carga la tarea de índice 2, con la cantidad de repuestos 2

    Reparacion1 M2("Ford","FIESTA","LOY457",9,20,2,2); //datos vehiculo.hora.min,
                                                    //indice de la tarea ,cant_rep

    cout<<M2;
    M2.Tarea(1,1); // Se carga la tarea de índice 1, con la cantidad de repuestos 1
    return 1;
}
```



```
PATENTE: ABX642
Marca:citroen
Modelo:C3
Hora Ingreso:8.45
Tarea realizada :C_Amortiguadores
Costo Parcial : 14000
COSTO ACUMULADO: 14000

PATENTE: ABX642
Tarea realizada :C_Amortiguadores
Cantidad : 1
valor parcial :3500
COSTO ACUMULADO: 17500

PATENTE: ABX642
Tarea realizada :Embrague
Cantidad : 2
valor parcial :17600
COSTO ACUMULADO: 35100

PATENTE: LOY457
Marca:Ford
Modelo:FIESTA
Hora Ingreso:9.20
Tarea realizada :Embrague
Costo Parcial : 17600
COSTO ACUMULADO: 17600

PATENTE: LOY457
Tarea realizada :Afinación
Cantidad : 1
valor parcial :5000
COSTO ACUMULADO: 22600
Press <RETURN> to close this window...
```

El código hágalo sobre Qt Creator. Si necesita enviar aclaraciones a su profesor, escribálas en un documento de texto. Comprima todos los archivos y súbalos al AV.