



“Universidad Autónoma del Estado de México”

Facultad de Ciencias

UA: Lenguajes de Programación

Periodo Escolar 2020B

Alumno: Francisco Javier de la Cruz Lugo.

Profesor: Dr. Erik Mendoza de la Luz

Título: “Ejercicios de Modularidad. Plantillas”



1. Escriba una plantilla de función llamada `maximo()` que devuelva el valor máximo de tres argumentos que se transmitan a la función cuando sea llamada.

```
template <class T>
T maximo(T x,T y,T z)
{
    T num_max;
    if (x >= y && x >= z)
        num_max = x;
    else if (y >= x && y >= z)
        num_max = y;
    else if (z >= x && z >= y)
        num_max = z;
    return num_max;
}
```

Esta plantilla asigna un tipo de dato **T** el cual va a adaptarse al tipo de dato entrante en la función, cabe mencionar que el tipo de dato que someterá estará declarado en la función principal. El argumento de la plantilla para **maximo()** consta de tres datos, mismos que son comparados dentro de la plantilla de la función cuando esta es invocada. El funcionamiento de esta se basa en la comparación de los mismos tres datos, de forma que la terminar de evaluarse la función esta designará un valor para el mayor dato de entres los tres asignados, en otras palabras, escoge el dato más grande y le asigna el nombre de la variable **num_max**.

2. Incluya la plantilla de función creada para el ejercicio anterior dentro de un programa en C++ completo que llame a la función con tres números enteros y luego con tres números de doble precisión.

Archivo adjunto: [programa#10.cpp](#)

3. Escriba una plantilla de función llamada `alcuadrado()` que calcule y devuelva el cuadrado del argumento único transmitido a la función cuando es llamada.

```
template <class T>
T alcuadrado(T num)
{
    T n_cuadrado;
    n_cuadrado = num * num;
    return n_cuadrado;
}
```

Esta plantilla asigna un tipo de dato **T** que se adaptará al tipo de dato entrante, este tipo de dato entrante estará declarado dentro de la función principal. El argumento de esta plantilla consta de un dato que será evaluado y operado dentro de la plantilla de la función, como se trata del cuadrado de un número, la función multiplica este numero por sí mismo y posteriormente lo guarda en la variable **n_cuadrado** para devolverla al final de esta.

4. Incluya la plantilla de la función creada en el ejercicio 3, dentro de un programa en C++ completo.

Archivo adjunto: [programa#11.cpp](#)