Exercise 5: Complex Numbers

This exercise requires you to use Class Template to define a class that returns the real and imaginary part of a Complex number

We'll cover the following

Problem Statment

In this exercise, you are required to use **Class templates**.

Make a **template** type class called `Complex`.

- Contains two `private` variables:
  - The **real** and **imaginary** element of a complex number.
- Has a constructor taking **real** and **imaginary** parts a *complex* number as input and initializing those.
- Has two other *member* functions:
  - `get_real()` : *returns* the **real** part of the *complex* number.
  - `get_im()` : *returns* the **imaginary** part of the *complex* number.

The code should be generalized hence should work for any data types.

> **Hint:** Use `template` type for this purpose.

Down below is what the *expected output* should look like.

Input :

```
int x=500, y=100;
```

Expected Output:

Expected Output:

```
Real part of the complex number is: 500
Imaginary part of the complex number is: 100
```

Write your code below. It is recommended that you try solving the exercise yourself before viewing the solution.

Good Luck!

Exercise | Solution

```
#include <iostream>
using namespace std;

template <class T>
class Complex {
private:
    T real, im;
public:
    Complex(T&, T&);
    T& get_real();
    T& get_im();
};

template <class T>
Complex<T>::Complex(T& r, T& i) {
    real = r;
    im = i;
}

template <class T>
T& Complex<T>::get_real() {
    return real;
}

template <class T>
T& Complex<T>::get_im() {
    return im;
}

int main()
{
    double x=500, y=100;
    Complex <double> comp(x,y);

    cout <<"Real part of the complex number is: " << comp.get_real()<<endl;
     cout << "Imaginary part of the complex number is: "<< comp.get_im() <<endl;
}
```