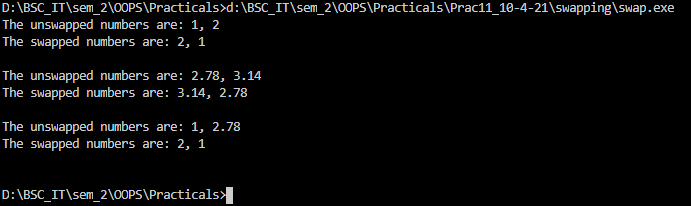
Practical 11: Templates

1. Aim: Using function template implement swapping of two variable using different data type.

Code:

|  |
| --- |
| // Using function template implement swapping of two variable using different data type.  #include<iostream>  using std::cout;  template<class num>void swap(num a, num b){      cout << "The unswapped numbers are: "<< a <<", "<<b<<"\n";      num c = a;      a = b;      b = c;      cout << "The swapped numbers are: "<< a <<", "<<b<<"\n\n";  }  template<class num1, class num2>void swap(num1 a, num2 b){      cout << "The unswapped numbers are: "<< a <<", "<<b<<"\n";      num2 c = a;      a = b;      b = c;      cout << "The swapped numbers are: "<< a <<", "<<b<<"\n\n";  }  int main(){      int i = 1, j =2;      float x = 2.78, y = 3.14;      swap(i, j);      swap(x, y);      swap(i, x);      return  0;  } |

Output:

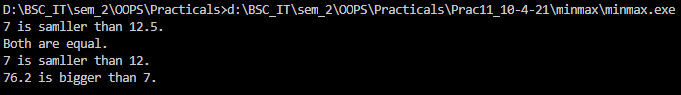


1. Aim: Using Class template find the max and min of two numbers using different data type

Code:

|  |
| --- |
| // Using Class template find the max and min of two numbers using different data type  #include<iostream>  #include<cmath>  using std::cout;  using std::cin;  template<class temp1, class temp2>  class num  {  private:      temp1 num1; temp2 num2;  public:      num(temp1 a, temp2 b){ num1 = a; num2 = b; }      void minMax(){          if (round(num1) > round(num2)){              cout << num2 <<" is samller than " << num1 <<".\n";          }else if (floor(num2) > floor(num1)){              cout << num2 <<" is bigger than " << num1 <<".\n";          }else{              cout << "Both are equal.\n";          }      }  };  int main(){      num<float, int>obj1(12.5, 7);      num<float, float>obj2(12.4, 12.6);      num<int, int>obj3(12, 7);      num<int, float>obj4(7, 76.2);      obj1.minMax();      obj2.minMax();      obj3.minMax();      obj4.minMax();      return 0;  } |

Output:



Write-Up:

