Harry (Templates and STL)

Template 01

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

template <class T>
class vector {
    public:
    T* arr;
    int size;
       vector(int m) {
        size=m;
        arr=new T[size];
       }
    T dotProduct(vector &v) {
        T d=0;
        for(int i=0;i<size;i++) {
            d += this->arr[i] * v.arr[i];
        }
        return d;
    }
};

int main()
{
    vector <float>v1(3);
    v1.arr[0] = 4.5;
    v1.arr[1] = 3.2;
    v1.arr[2] = 1.8;
    vector <float>v2(3);

    v2.arr[0]= 1.4;
    v2.arr[1]=0.5;
    v2.arr[2]=1.7;
    float a= v1.dotProduct(v2);
    cout<<a<<endl;

    return 0;
}
```

Template Multiple parameter

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

template <class T1,class T2>
```

```
class myClass {
    public:
    T1 data1;
    T2 data2;
    myClass(T1 a, T2 b) {
        data1=a;
        data2=b;
    }
    void display() {
        cout<<this->data1<<endl<<this->data2;
    }
};

int main()
{
    myClass<char, float> obj('c',1.8);
    myClass<int, double>obj1(12,5.6);
    myClass<int,char>obj2(7,'a');
    obj.display();
    obj1.display();
    obj2.display();
    return 0;
}
```

Default parameter

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

template <class T1=int, class T2=float, class T3=char>
class Billy{
    public:
    T1 a;
    T2 b;
    T3 c;
    Billy(T1 x,T2 y,T3 z) {
        a=x;
        b=y;
        c=z;
    }
    void display() {
        cout<<"The Value of a is: "<<a<<endl;
        cout<<"The Value of b is: "<<b<<endl;
        cout<<"The Value of c is: "<<c<<endl;
    }
};
```

```
int main()
{
    Billy<> ob1(4, 6.4, 'c');   // Default
    ob1.display();
    cout<<endl;
    Billy<float, char, char> ob2(1.6, 'a', 'b');
    ob2.display();
    return 0;
}
```

Template Function

```
#include<iostream>
#include <cmath>
using namespace std;
template<class T1, class T2>
float funcAverage(T1 a, T2 b){
    float avg = (a+b)/2.0;
    return avg;
}

template<class T>
void swapp(T &a,T &b) {
    T temp=a;
    a=b;
    b=temp;
}

int main()
{
    float a;
    a = funcAverage(3,2);
    //cout<<"The average is: "<<fixed<<setprecision(3)<<a;
    cout<<"The average is: "<<a<<endl;
    int x=6,y=99;
    swapp(x,y);
    cout<<x<<" "<<y;
    return 0;
}
```

STL

Vector 01

```
#include<iostream>
#include<vector>
```

```
using namespace std;

void display(vector<int> &v) {
    for(int i=0;i<v.size();i++) {
        cout<<v[i]<<" ";
        cout<<v.at(i)<<" ";  // It tells the element present at ith position
    }
    cout<<v.at(1)<<" ";
    cout<<v.at(0)<<" ";
    cout<<endl;
}

int main()
{
    vector<int> vec;  //Zero length vector
    int elem, size;
    cout<<"Enter the size of vector: ";
    cin>>size;
    for(int i=0;i<size;i++) {
        cout<<"Enter the elemnet of vector: ";
        cin>>elem;
        vec.push_back(elem);  // Insert at the end of vector
    }

    //vec.pop_back();   // Delete the elast element of vector

    display(vec);

    vector<int> :: iterator iter = vec.begin();  //Creating an iterator
    //vec.insert(iter, 566);   //Insert at beginning
    //vec.insert(iter+1, 78);   //Insert on second place
    //vec.insert(iter+1, 5, 145);  // Five copy of 145 inserted from second
place
    display(vec);
    return 0;
}
```

Vector 02 with template

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

template <class T>
void display(vector<T> &v) {
    cout<<"Displaying the vector"<<endl;
```

```
    for(int i=0;i<v.size();i++) {
        cout<<v[i]<<" ";
        //cout<<v.at(i)<<" ";  // It tells the element present at ith position
    }
    cout<<endl;
}

int main()
{
    vector<int> vec1; //Zeor length vector
    display(vec1);

    vector<char> vec2(4);  //4 element character vector
    vec2.push_back('g');
    display(vec2);

    vector<char> vec3(vec2);  // 4 element character vector from vec2
    display(vec3);

    vector<int> vec4(6,10);  // 6-element vector of 10
    display(vec4);
    cout<<vec4.size();
    return 0;
}
```

List

```
#include<iostream>
#include<list>

using namespace std;

void display(list<int> &lst) {
    list<int> :: iterator it;
    for(it = lst.begin(); it != lst.end(); it++) {
        cout<<*it<<" ";
    }
    cout<<endl;
}
int main()
{
    list<int> list1;  //List of zero length

    list1.push_back(5);
    list1.push_back(7);
    list1.push_back(9);
    list1.push_back(12);
    list1.push_back(9);
```

```
    list<int> :: iterator iter1;
    iter1 = list1.begin();  //iter is the reference of the first element of
list1
    cout<<"\nElements of list1: ";
    cout<< *iter1<<" ";
    iter1++;
    cout<< *iter1<<" ";
    iter1++;
    cout<< *iter1<<" ";
    iter1++;
    cout<< *iter1<<" ";

    cout<<"\nDisplaying list 1 by display() function: ";
    display(list1);

    //Removing elements from list

    //list1.pop_back();  //Deleted the last element
    //list1.pop_back();  //Deleted the last element

    //list1.pop_front();  //Deleted the element from beginning

    //list1.remove(9);  //Delete a particular element which value is given in
function as parameter

    //Sorting the list
    //list1.sort();

    display(list1);

    list<int> list2(3);  // Empty list of size 3
    list<int> :: iterator iter2;
    iter2 = list2.begin();
    *iter2 = 54;
    iter2++;
    *iter2 = 6;
    iter2++;
    *iter2 = 9;
    iter2++;

    cout<<"\nElements of list2: ";
    display(list2);

    //Merging the two lists
    list1.merge(list2);  //Merging the list2 into list1
    cout<<"\nList 1 after merging: ";
    display(list1);
```

```
    //Reversin the list
    list1.reverse();
    cout<<"\nList 1 after reversing: ";
    display(list1);

    return 0;
}
```

Map

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

//Map is an associative array

int main()
{
    map<string, int> marksMap;
    marksMap["Azam"] = 98;
    marksMap["Armaan"] = 85;
    marksMap["Arbaz"] = 90;

    marksMap.insert({{"Ilaix",180},{"Pedri", 8}});
    map<string, int> :: iterator iter;
    for(iter= marksMap.begin(); iter!=marksMap.end(); iter++) {
        cout<< (*iter).first<<" "<<(*iter).second<<endl;  // Printing the pair
    }

    cout<<"The size is: "<<marksMap.size()<<endl;
    cout<<"The maiximum size is: "<<marksMap.max_size()<<endl;
    cout<<"The Empty's return value is: "<<marksMap.empty()<<endl;

    return 0;
}
```

Function Objects (Functor)

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

//Function Objects: Function wrapped in a class so that it available like an
object (It is also called Functor)
int main()
{
```

```
    int arr[] = {1,33,14,120,54,77};
    //sort(arr,arr+5);  //first five element will be sorted in arr[] array (By
default it sort in ascending order)
    sort(arr,arr+6, greater<int>());   // First six elements will be sorted in
descending order
    for(int i=0;i<6;i++) {
        cout<<arr[i]<<endl;
    }

    return 0;
}
```