# LAB 7

1. /*Write a program to find the maximum of two numbers using function template.*/

```
#include<iostream>
using namespace std;
template<class T>
T maxx(T a, T b)
{
   T max;
   if(a>b)
   {
      max =  a;
   }
   else
      max = b;
   return max;
}
int main()
{
   int x, y;
   cout<<"Enter any two number "<<endl;
   cin>>x>>y;
   cout<<"The max number is "<<maxx(x,y)<<endl;
}
```

2. /*Write a program to find the minimum of two numbers using function template.*/

```
#include<iostream>
using namespace std;
template<class T>
T minn(T a, T b)
{
   T min;
   if (a<b)
   {
      min = a;
   }
   else
   {
      min = b;
   }
   return min;
}
int main()
{
   int x,  y;
   cout<<"Enter any two number "<<endl;
   cin>>x>>y;
   cout<<"The minimum number is "<<minn(x,y)<<endl;
}
```

3. /*Write a program to find the maximum of  two data members of a classes using the concept of a template.*/

```
#include<iostream>
using namespace std;
template<class T>
class maxx
{
   T a, b;
   public:
   maxx(T x, T y)
   {
      a = x;
      b = y;
   }
   T get_max()
   {
      T m;
      m = (a>b)?a:b;
      return m;
   }
};
int main()
{
   int p,q;
   cout<<"Enter any two numbers "<<endl;
   cin>>p>>q;
```

```
   maxx<int>o(p,q);
   cout<<"The maximum number is "<<o.get_max()<<endl;
}
```

4. /*Write function template for finding the minimum value contained in an array.*/

```
#include<iostream>
using namespace std;
template<class T>
T maxx(T a[], int n)
{
   T max = a[0];
   for (int i = 0; i < n; i++)
   {
      if(max>a[i])
      {
         max =  a[i];
      }
   }
   return max;
}

int main()
{
   int n, x[100];
   cout<<"Enter the size of array "<<endl;
```

```
   cin>>n;
   cout<<"Enter the element of array "<<endl;
   for (int i = 0; i < n; i++)
   {
      cin>>x[i];
   }
   cout<<"The minimum number is "<<maxx(x,n)<<endl;
}
```

5. /*Write a program to read two numbers from the user then divide first number by second number only if second number is not zero. If second number is zero then throw divide by zero exception and handle it.*/

```
#include<iostream>
using namespace std;
int main()
{
   int a, b;
   cout<<"Enter any two number "<<endl;
   cin>>a>>b;
   try
   {
      if(b==0)
      {
         throw b;
      }
      else
```

```
    {
        cout<<"Result is "<<(float)a/b<<endl;
    }
  }
  catch(int e)
  {
    cout<<"Divide by zero is not possible b = "<<e<<endl;
  }
}
```

6. /*Customize the above program so that throw point outside the try block (i.e., used function to divide and call this function from try block).*/

```
#include<iostream>
using namespace std;
void divide(int a, int b)
{
  if (b==0)
  {
    throw b;
  }
  else
  {
    cout<<"Result is "<<(float)a/b<<endl;
  }
}
```

```
int main()
{
   int a, b;
   cout<<"Enter any two number "<<endl;
   cin>>a>>b;
   try
   {
      divide(a, b);
   }
   catch(int e)
   {
      cerr<<"Divide by zero is not possible b = "<<e<<endl;
   }
}
```

7. /*Write a program to read different value of a variable such as 0, 1,-1, etc .Throw and catch the different types of exceptions depend on the value of a variable such as if 0 then throw integer exception, if 1 throw character exception and so on ( note: use multiple catch blocks).*/

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

void num(int a)
{
   if(a==1)
   {
```

```
        throw "a";
    }
    else if (a==0)
    {
        throw a;
    }
    else if (a==-1)
    {
        throw 5.5;
    }
    else
    {
        cout<<"The input is invalid "<<endl;
    }
}

int main()
{
    int a;
    cout<<"Enter -1, 0 or 1 "<<endl;
    cin>>a;
    try
    {
        num(a);
    }
    catch(char const* e)
```

```
    {
       cout<<"The character is "<<e<<endl;
    }
    catch(int e)
    {
       cout<<"The integer is "<<e<<endl;
    }
    catch(double e)
    {
       cout<<"The double is "<<e<<endl;
    }
}
```