**Software Development Lab – II [15B17CI271]**

**Assignment Sheet**

**Week 10**

|  |  |  |
| --- | --- | --- |
| **COURSE OUTCOMES** | | **COGNITIVE LEVELS** |
| C173.1 | Write programs in C++ to implement OOPs concepts related to objects, classes, constructor, destructor, and friend function. | Apply Level (Level 3) |
| **C173.2** | Write programs in C++ using OOPs concept like encapsulation, inheritance, polymorphism and abstraction. | Apply Level (Level 3) |
| **C173.3** | Write programs in C++ using Standard Template Library. | Apply Level (Level 3) |
| **C173.4** | Perform exception handling in C++ programs. | Apply Level (Level 3) |
| **C173.5** | Write MySQL queries to perform operations like ADD, DELETE, UPDATE, SELECT on relational databases. | Apply Level (Level 3) |

Upload the Solution with output as single pdf file in the given link:

<https://forms.gle/d69aRcL18yc9WCf96>

(Naming convention: **Batch\_Rollno\_Assignmentno e.g., B1\_198276354\_A1)**

**Topics: Function Templates, Class Templates and STL**

**Q1.** Predict the output of following program.

template <typename T>

void abcd(const T&x)

{    static int count = 0;

    cout << "x = " << x << " count = " << count << endl;

    ++count;

    return;

}

int main()

{

    abcd<int> (1);

    cout << endl;

    abcd<int>(1);

    cout << endl;

    abcd<double>(1.1);

    cout << endl;

    return 0;

}

**Q2.** Predict the output of the following program.

|  |  |
| --- | --- |
| template <class T>  class Test  { private:      T val;  public:      static int count;      Test()  {   count++;   }  };  template<class T>  int Test<T>::count = 0; | int main()  {      Test<int> a;      Test<int> b;      Test<double> c;      cout << Test<int>::count   << endl;      cout << Test<double>::count << endl;      return 0;  } |

**Q3.** Create a template class calculator to perform addition, subtraction, multiplication and division of two numbers. Show the results for different datatypes.

**Q4.** Write templates for the two functions, namely minimum and maximum. Minimum function should accept two arguments and return the value of the arguments that is the lesser among the two. Maximum function should accept two arguments and return the value of the arguments that is the greater among the two values. Design a simple driver program that demonstrates the templates with various data types.

**Q5. C**reate your own template class MyVector with data members and member functions such that the size(), push\_back() and pop\_back() functionalities of Vector can also be performed by MyVector.

**Q6.** Implement the following problem using vector STL in c++.

Consider two arrays of similar type, having different numbers of elements. Take the array values from user till a negative value for both the arrays. Insert those elements of second array into first array which are not present in the first array. Also display the first array after insertion in sorted form.