SRN | | | | | | | | | | | | |

PES

PES INSTITUTE OF TECHNOLOGY, Bengaluru

(Autonomous Institute under VTU, Belgaum and UGC, New Delhi)

14CS337
CS/IS

JAN – MAY- 2017: SEMESTER END EVALUATION (SEE) B.TECH. VI SEMESTER

14CS337 – GENERIC PROGRAMMING-THEORY PAPER

Instructions: You may bring any written matter. You cannot have any printed matter. You cannot exchange the any material during the test.

Time: 2.5 Hr | Answer All Questions. Neglect syntax errors | Max Marks: 80

1.	a)	Identify the sources of inefficiency. Rewrite the code supporting the same semantics. You may indicate what the Employee class should support to make some of these operations efficient. Give your reasons.
		string FindAddr(list<Employee> emps, string name)
		{
		for(list<Employee>::iterator i = emps.begin();
		i != emps.end();
		i++)
		{
		if(*i == name)
		{
		return i->addr;
		}
		}
		return "";
		}
			6

	b)	template <typename T>
		void fun(const T&x)
		{
		static int count = 0;
		cout << "x = " << x << " count = " << count << endl;
		++count;
		return;
		}
		
		fun<int> (1);
		cout << endl;
		fun<int>(1);
		cout << endl;
		fun<double>(1.1);
		cout << endl;
		Find the output. Give your reasons.
			3+3

	c)	) template <class T, int max>
		int arrMin(T arr[], int n)
		{
		int m = max;
		for (int i = 0; i < n; i++)
		if (arr[i] < m)
		m = arr[i];
			3+3

SRN | | | | | | | | | | | | |

```
  return m;
}
```

What does the function do?

```
int a[] = {20, 15, 10, 30};
```

What happens in the following cases? Why?

```
        cout << arrMin<int, 100>(a, 4) << "\n";
        cout << arrMin<int, 0>(a, 4) << "\n";
```

2. a) Find the output in each case. Justify your answer in a single sentence. 6+6

```
template <class T>
T mymax (const T &a, const T &b)
{
        cout << "one ";
   return (a > b)? a : b;
}

template <>
int mymax <int> (const int &a, const int &b)
{
   cout << "two ";
   return (a > b)? a : b;
}

template <>
long mymax <long> (const long &a, const long &b)
{
   cout << "three ";
   return (a > b)? a : b;
}

template <>
string mymax(const string &a, const string &b)
{
   cout << "four ";
        return (a > b)? a : b;
}
int main ()
{
   int a = 10, b = 20;
        string c = "apple";
        string d = "banana";
   cout << mymax <int> (a, b) << "\n";
        cout << mymax (10l, 20l) << "\n";
        cout << mymax <long> (a, b) << "\n";
        cout << mymax <char> (c[0], d[0]) << "\n";
        cout << mymax(c, d) << "\n";
        cout << mymax(c.c_str(), d.c_str()) << "\n";
}
```

b) Find the output. Please note that default parameters are allowed for template functions. 6

```
template <typename T = int, int count = 3>
```

SRN | | | | | | | | | | | |

```
T foo(T x)
{
        for(int ii = 0; ii < count; ii++)
        {
                x = x * x;
        }
        return x;
};

int main()
{
        int x(2);
        cout << x  << ": " << foo<>(x) << endl;
        cout << x  << ": " << foo<int, 2>(x) << endl;
        cout << 2.5  << ": " << foo<double, 2>(2.5) << endl;
}
```

c) Find the output. 6

```
int main ()
{
   vector<int> myvector;
   for (int i = 1; i <= 10; i++)
   myvector.push_back(i);

   myvector.erase (myvector.begin() + 6);
   for (int i = 0; i < myvector.size(); ++i)
    cout << ' ' << myvector[i];
    cout << "\n";
   myvector.erase (myvector.begin(), myvector.begin() + 4);
   for (int i = 0; i < myvector.size(); ++i)
    cout << ' ' << myvector[i];
    cout << "\n";
   myvector.erase (--myvector.end());
   for (unsigned i = 0; i < myvector.size(); ++i)
    cout << ' ' << myvector[i];
    cout << "\n";
}
```

3. a) Find the output. 6

```
int main ()
{
     vector<int> myvector(10);
     iota(begin(myvector), end(myvector), 1);
         sort(begin(myvector), end(myvector), greater<int>());
         sort(begin(myvector) + myvector.size() / 2, end(myvector),  less<int>());
         for_each(begin(myvector), end(myvector), [](auto e){cout << e << "\t"; });
         cout << "\n";
         cout << *find_if(begin(myvector), end(myvector),
         [myvector](auto e) { return e < myvector.size() / 2; }) << "\n";
         cout << count_if(begin(myvector), end(myvector),    [myvector](auto e) { return e <
myvector.size() / 2; }) << "\n";
}
```

SRN | | | | | | | | | | | | |

	b)	Find the output.
		using namespace std;
		struct what
		{
		bool operator()(int x, int y)
		{
		return x/10 > y/10;
		}
		};
		template<typename C>
		void disp(C s)
		{
		for(auto e : s) cout << e << "\t";
		cout << "\n";
		}
		int main ()
		{
		int a[] = { 1, 3, 5, 7, 9, 11, 13, 15, 17, 19};
		set<int> s1(begin(a), end(a));
		set<int, greater<int> > s2(begin(a), begin(a) + 5);
		set<int, what> s3(begin(a), end(a));;
		disp(s1); disp(s2); disp(s3);
		}
			6

	c)	Find the output.
		
		template<int X, int N>
		struct foo
		{
		enum { res = X * foo<X, N - 1>::res };
		};
		template<int X>
		struct foo<X, 0>
		{
		enum { res = 1 };
		};
		
		int main()
		{
		cout << foo<5, 3>::res << '\n';
		cout << foo<5, 0>::res << '\n';
		cout << foo<10, 1>::res << '\n';
		}
			6

4	a)	iterator_traits has the following member types. difference_type value_type pointer reference iterator_category input_iterator_tag output_iterator_tag	10

SRN | | | | | | | | | | | |

		forward_iterator_tag bidirectional_iterator_tag random_access_iterator_tag Given an iterator, write code for the following. i) Create a variable of the value type ii) Find the iterator category and display a string indicating the category iii) Create a variable to hold the difference between two such iterators. iv) Write a swap function which takes a pair of iterators and swaps the corresponding value. { hint: how to create local variable of the value type? }	
	b)	Answer these questions precisely. i) Why generic class in Java cannot have static members? ii) if A extends B, does that mean a bag of A extends a bag of B? Give reasons. iii) Why we can instantiate the generic type in a generic class or a generic function? iv) Why does begin(s) where is of set type in C++, return a const_iterator? v) Why we can implement a queue using a vector by composition?	10