Name: Atul Kumar Reg. No: 12102801
Roll No: RD2110B79 Assignment: 03
Course Code: CAP444 SET - "D"

Q1. How class templates can be used with inheritance? Explain the concept with a suitable example.

Ans: It is possible to inherit from a template class. All the usual rules for inheritance and polymorphism apply. If we want the new, derived class to be generic it should also be a template class; and pass its template parameters along to the base class.

Ex:
```
#include <iostream>
using namespace std;
template<typename T>
class inherit
{
  public: T a,b;
void show()
{
```

```
cout<<"Enter a & b value"<<endl;
 cin>>a>>b;
};
template<typename T1>
class inherit1 : public inherit<T1>
{
 public:
void display()
{
inherit<int> ob;
 ob.show();
cout<<ob.a<< ob.b;
};
 int main()
inherit1 <int> ob;
ob.display();
return 0;
}
```

Output

```
Enter a & b value
12
12
1212
```

Q2: Why overloading the function template is important? Explain with example.

Ans: Overloading the function template is important because it improves code readability and allow code reusability. It also save memory space, consistency and readability. It speeds up the execution of the program and code maintenance also become easy. Function overloading in template achieved by function template either by a non-template function or by another function template.

Ex:
```
# include <iostream>
using namespace std;
template <class T>
void f (T x, T y)
{
```

```
cout<<"Template"<<endl;
}
void f(int w, int z)
{
cout<<"Non-template"<<endl;
}

int main()
{
   f(1, 2);
   f('a', 'b');
   f(1, 'b');
}
```

Output

```
Non-template
Template
Non-template
```

Q3. How multiple exceptions can be controlled? Explain with example.

Ans: Multiple exceptions can be controlled by using multiple catch block in a program

(1) Start the program
(2) Declare and define the function test()
(3) Within try block check whether the value is greater than zero or not
(4) If the value greater than zero throw the value and catch the corresponding exception otherwise throw the character and catch the corresponding exception.
(5) Read the integer and character values for the function testV
(6) Stop the program.

Eg.
```
#include<iostream>
using namespace std;
void test(int x){
  try{
    if(x>0)
      throw x;
    else
      throw 'x';
  }
  catch(int x){
cout<<"catch a integer and that
integer is:"<<x<<endl;}
  catch(char x){
cout<<" catch a character and that
character is:"<<x<<endl;}
}
int main(){
  cout<<"Testing multiple catches"<<endl;
  test(10);
  test(0);
  return 0;}
```

Output

Testing multiple catches
catch a integer and that integer is: 10
catch a character and that character is: x