

Software Development Lab – II [15B17CI271]

Assignment Sheet

Week 8

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)

Note: Students are advised to submit their solutions to respective lab faculty. The solution file must be named as “rollno_first name_w8.doc” (here w8 represents week 8).

1. Write a program to implement the exception handling with multiple catch Statements. You have to take three input values from user as Integer value, Character value and double value. For each above values you'll have to handle it with the catch statement.
2. What output is produced by the following code?

```
int wait_time = 46;

try
{
    cout << "Try block entered.\n";

    if (wait_time > 30)
        throw wait_time;

    cout << "Leaving try block.\n";
}

catch(int thrown_value)
{
    cout << "Exception thrown with\n"
    << "wait_time equal to " << thrown_value << endl;
}

cout << "After catch block." << endl;
```

3. What would be the output produced by the code in Question 2 if we make the following change? Change the line

```
int wait_time = 46;

to
```

```
int wait_time = 12;
```

4. What happens when a throw statement is executed? This is a general question. Tell what happens in general, not simply what happens in the code in Question 2 or some other sample code.
5. What is the output produced by the following program?

```
#include <iostream>

using namespace std;

void sample_function(double test) throw (int);

int main()
{
    try
    {
        cout << "Trying.\n";
        sample_function(98.6);
        cout << "Trying after call.\n";
    }
    catch(int)
    {
        cout << "Catching.\n";
    }
    cout << "End of program.\n";
    return 0;
}

void sample_function(double test) throw (int)
{
    cout << "Starting sample_function.\n";
    if (test < 100)
        throw 42;
}
```

6. What is the output produced by the program in Question 5 if the following change were made to the program?

Change `sample_function(98.6);`
in the try block to
`sample_function(212);`

7. What happens when an exception is never caught?
8. Can you nest a try block inside another try block?
9. What will be the output of the following code snippet?

```
void myFunction(int test) {  
    try{  
        if (test)  
            throw test;  
        else  
            throw "Value is zero";  
        }  
    catch (int i) {  
        cout << "CaughtOne " ;  
    }  
    catch (const char *str) {  
        cout << "CaughtString " ;  
    }  
}  
  
int main() {  
  
    myFunction(1);  
    myFunction(2);  
    myFunction(0);  
    myFunction(3);  
    return 0;  
}
```

- a) CaughtOne CaughtOne CaughtString CaughtString
- b) CaughtOne CaughtString CaughtString CaughtOne
- c) CaughtOne CaughtOne CaughtOne CaughtOne
- d) CaughtOne CaughtOne CaughtString CaughtOne

10. What will be the output of the following code snippet?

```
#include<iostream>  
using namespace std;  
struct MyException : public exception {  
    const char * what () const throw () {  
        return "C++ Exception";  
    }  
};  
  
int main() {  
    try {  
        throw MyException();  
    }catch(MyException& e) {  
        std::cout << "MyException caught" << std::endl;  
        std::cout << e.what() << std::endl;  
    } catch(std::exception& e) {  
        std::cout << "Exception caught" << std::endl;  
        std::cout << e.what() << std::endl;  
    }  
}
```

- a) MyException caught
C++ Exception
- b) C++ Exception

MyException caught
c) Exception caught
C++ Exception
d) C++ Exception
MyException caught
Exception caught