**Lab – 10**

Exception handling – try, catch & throw

1. Run the following code to understand why exception handling is required –

// Without exception handling

float division(int x, int y){

return (x/y);

}

int main(){

int i=50, j=0;

float k=0;

k=division(i,j);

cout<<k<<endl;

}

1. Run and understand the following program which uses try-throw-catch –

float division(int x, int y) {

if( y == 0 ) throw “Custom msg - divide by zero!";

return (x/y);

}

int main () {

int i = 25,j = 0;

float k = 0;

try {

k = division(i, j);

cout << k << endl;

} catch (const char\* e) { cout<< e << endl;}

}

1. Complete the following code by writing multiple catch statements after try block –

float division(int x, int y) {

//...

}

int main () {

float k = 0;

try {

k = division(25, k);

cout << k << endl;

}

// first catch block …

// second catch block …

}

1. Write a simple program to implement default catch(…).
2. Practice to throw an integer, char, double, char array and string exceptions.
3. Write a simple program to illustrate exception specification, which means putting a constraint on the type of exception that a function can throw.
4. Why just the *if*-statements can’t be used instead of exception handling mechanism? Discuss it with other students and the lab instructor.