

# CS3201: OBJECT ORIENTED PROGRAMMING LABORATORY

*Topic: Exceptions, Revision Exercises, Inline functions*

**Lab: 08**

**Date: 10/05/2024**

## OBSERVATION

1. Which of the following are advantages of exception handling?
  - a) Remove error-handling code from the software's main line of code.
  - b) A method writer can choose to handle certain exceptions and delegate others to the caller.
  - c) An exception that occurs in a function can be handled anywhere in the function call stack.

2. Predict the output-

```
#include <iostream>
using namespace std;
int main()
{
    try
    {
        throw 10;
    }
    catch (...)
    {
        cout << "default exception\n\n";
    }
    catch (int param)
    {
        cout << "int exception\n\n";
    }
    return 0;
}
```

3. Is this code correct? Justify.

```
#include <iostream>
using namespace std;
int fun(int x, int y = 0, int z = 0)
{ return (x + y + z); }
int main()
{
    cout << fun(10);
    return 0;
}
```

4. What is the output? Discuss ways to fix this

```
#include<iostream>
#include<stdlib.h>
using namespace std;
class Test
{
public:
    Test()
    { cout << \"Constructor called\\n\"; }
};
int main()
{
    Test *t = (Test *) malloc(sizeof(Test));
    return 0;
}
```

5. When the function is small, what do we replace the macros with?