Data Structures and Object Oriented Programming

Lecture 26

Dr. Naveed Anwar Bhatti

Webpage: naveedanwarbhatti.github.io

Error Handling

Techniques for Error Handling

- Abnormal termination
- Graceful termination
- Return error code from a function
- Exception handling

Example – Abnormal Termination

```
void GetNumbers(int& a, int& b)
{
    cout << "Enter two integers";
    cin >> a >> b;
}


int Quotient(int a, int b)
{
    return a / b;
}


void OutputQuotient(int quo)
{
    cout << "Quotient is " << quo << endl;
}
```

```
int main()
{
    int a, b, quot;
    for (int i = 0; i < 10; i++)
    {
        GetNumbers(a, b);
        quot = Quotient(a, b);
        OutputQuotient(quot);
    }
    return 0;
}
```

Output

```
Enter two integers
10
10
Quotient is 1
Enter two integers
10
0
Program terminated abnormally
```

Graceful Termination

- Program can be designed in such a way that instead of abnormal termination, that causes the wastage of resources, program performs clean up tasks

Example – Graceful Termination

```
void GetNumbers(int& a, int& b) {
    cout << "Enter two integers";
    cin >> a >> b;
}

int Quotient(int a, int b) {
    if (b == 0)
    {
        cout << "Denominator can’t be zero" << endl;
        exit(1);
    }
    return a / b;
}

void OutputQuotient(int a, int b, int quo) {
cout << "Quotient is " << quo << endl;
}
```

```
int main() {
    int a, b, quot;
    for (int i = 0; i < 10; i++)
    {
        GetNumbers(a, b);
        quot = Quotient(a, b);
        OutputQuotient(a, b, quot);
    }
    return 0;
}
```

Output

```
Enter two integers
10
10
Quotient is 1
Enter two integers
10
0
Denominator can’t be zero
```

Return Error Code

- Programmer has avoided the system crash but the program is no longer running.
- Solution-> **Return Error Code**

Example – Return Error Code

```
void GetNumbers(int& a, int& b) {
    cout << "Enter two integers";
    cin >> a >> b;
}

bool Quotient(int a, int b, int& retVal) {
    if (b == 0) {
        return false;
    }
    retVal = a / b;
    return true;
}

void OutputQuotient(int a, int b, int quo) {
    cout << "Quotient is " << quo << endl;
}
```

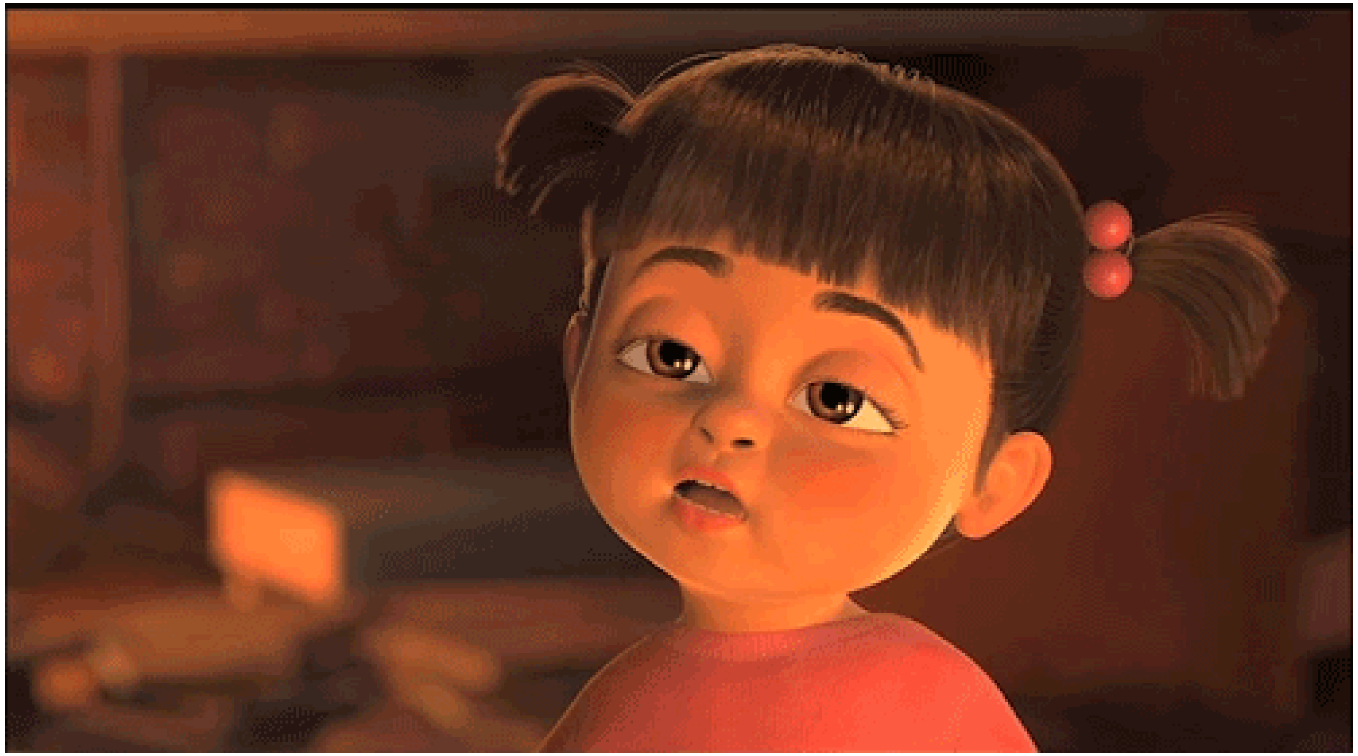
```
int main() {
    int a, b, quot;
    for (int i = 0; i < 10; i++)
    {
        GetNumbers(a, b);
        while (!Quotient(a, b, quot)) {
            cout << “Denominator can’t be zero. Give
            input again";
            GetNumbers(a, b);
        }

        OutputQuotient(a, b, quot);
    }
    return 0;
}
```

Output

```
Enter two integers
10
0
Denominator can’t be zero. Give input again
Enter two integers
10
10
Quotient is 1
```

Thanks a lot



If you are taking a Nap, **wake up**.......Lecture Over