Exception Handling

Contents

- Types of errors
- C way error handling
- Need of exception handling
- Exception handling mechanism.
- How to implement exception handling

Types of errors

- Syntax or compile time error.
- Run time error
- Logical error
- Linker error.

- But In exception handling we have to consider run time errors.

C way of error handling

- Run time errors are handled in C programming by using if – else and switch case statement.
- Problems wit C style error handling.

    1. The error message displayed is not friendly to user.

    2. Business logic and error logic is not separate.

    3. Error code that is returned needs to check again.

C style error handling

For example

```
int pop()
{
    if( top== -1 )
    {
        printf("Stack is empty");
        return -9999;
    }
}
```

Error Checking

Business Logic

```
int main()
{
        int  val  =  pop ( );
}
```

Return value needs to be checked again

Exception

- Exception represents run time error.
- **For example**
  - Division by zero.
  - Stack overflow
  - Memory allocation failed.
  - Uninitialized variable is used

Identify block where error may be occur

- By using try block we can identify block where error may be occur.

- **For e.g**

```
try
{
        code……
}
```

Handle error

- Error is handled by catch block.
- **For e.g.**

```
catch(.....)
{
        ..................
}
```

- Catch block must represented below try block.

throw error message

- Error message generated by using throw keyword.
- **For e.g.**

```
try
{
        throw “error occurred”;
}
catch(char * msg)
{
        cout<< msg;
}
```

Represent multiple catch block

- We can represent multiple catch block.

For e.g.

```
try
{
        throw -9999;   //error value
}
catch(char *msg)
{
        ……….
}
catch(int val)
{
        ……….
}
```

```
catch(float  val)
{
        ……….
}
```

If exception is not handled

- If exception is not handled then we can use ellipse catch block.
- It will handle all exceptions.

For e.g.

```
try
{
        ……….
}
catch(int val)
{
        ……….
}
catch( . . .)    //ellipse catch block
{
        ……….
}
```

Lab Assignments

- Write a program to perform stack operation and give proper error message by using exception handling.
    1. Stack is full
    2. Stack is empty
- Write a program to perform division operation and if user enters denominator value zero then throw error message.