COMPILER DESIGN PROJECT

TRY CATCH PARSER

Try Catch and Finally

Try - The try statement allows you to define a block of code to be tested for errors while it is being executed.

Catch - The catch statement allows you to define a block of code to be executed, if an error occurs in the try block.

Finally - The finally statement lets you execute code, after try...catch, regardless of the result

Syntax

```
void SomeMethod(Params)
{

    ------------//normal statements
    try
    {
        ------------
        ------------//exception causing and its related statements
    }
    catch(SomeException1 | SomeException2 e)
    {
        ------------//statements that takes proper actions
        ------------//SomeException
    }
    catch(SomeException3 e)
    {
        ------------//statements that takes proper actions
        ------------//SomeException
    }
    finally
    {
        ------------//cleanup code
    }
    ------------//normal statements

}
```

Handled

- Single line comment.
- Multiline comment.
- A function with return type and can have zero or more arguments.
- Simple declaration, assignment or unary operations.

```
/*Multiline Comment Supported


Simple function with zero or more arguments */
int fun(int a, String s){

    //Body can contain simple assignment
    int a=10;

    //Body can contain increment operation
    a++;
    --a;

}
```

- Try should be inside the function body.
- Multiple occurrences of try ,catch and finally.
- Body of try, catch and finally can be empty.
- Catch should have one or more arguments.
- Try-catch or try-finally can occur without finally and catch respectively.

```
int fun(int a){

    try{}
    finally{}

    try{}
    catch(NullPointerException | SQLException ex){}
    catch(ArithematiException ex){}

    try{}
    catch(ArithematiException ex){}
    finally{}
}
```

- Nested try,catch and finally can occur.
- Try, catch and finally block can occur anywhere with or without expressions.

```
int fun(int a){
    int z;
    try{
        int p;
        try{}
        finally{}
    }
    catch(exception e){
        z = 90;
        try{ z++;}
        catch(exception e){}
        catch(exception e){--z;}
        finally{ z=0; }
    }
    finally{}

    int a=16;
    try{}
    catch(exception e){}
    finally{}

    a++;
}
```

- Catch or finally cannot occur without try.
- Catch Should have atleast single parameter.

```
int fun(int a){
    //error
    catch(exception e){}

    //catch after finally error
    try{}
    finally{}
    catch{}

    try{
        //error
        finally{}
    }
}
```

- Alone try cannot occur

```
int fun(int a){

    try{}

}
```

Not-Handled

- Our program can only handle simple declaration, assignment and unary syntax only rest is not handled.
- Function without body is not handled.
- Try with resource is not handled.
- Keyword throw, throws not handled.