

# Lecture 7

## Classes and Exceptions

Richard J. Orgass  
Heinz School  
Carnegie Mellon University

Carnegie Mellon

1

## Agenda

---

- Classes
- Exceptions

Carnegie Mellon

2

## Classes

---

- Class declarations
  - public attributes
  - private attributes
  - data attributes
  - method attributes
  - inner classes
- Method definitions
  - :: notation
  - method signatures in definition match declaration
  - definitions have code for the methods

Carnegie Mellon

3

## Class Declaration

```
class Stack {  
public:  
    class Full{};  
    class Empty{};  
  
    void push(int i) throw(Full);  
    int pop() throw(Empty);  
  
private:  
    int *store;  
    int height = 0;  
    int maxHeight;  
}
```

class name

public innter class declarations

public methods

private data

Carnegie Mellon

4

## Method Definitions

```
Stack :: push(int i) throw(Full) {  
    if (height >= maxHeight) throw(new Full());  
    store[height] = i;  
    height ++;  
}
```

Class name

Method Name

throw statement

object to be thrown

Carnegie Mellon

5

## Exceptions

- A C++ entity of any kind can be thrown
  - string
  - integer
  - class
- Exceptions are handled by the type that was thrown
  - catch( char \*s) {...} for strings
  - catch(Full){...} for an object of type Full
  - etc
- We've seen a throw example

Carnegie Mellon

6

## Catching an Exception

---

```
int main(int argc, char **argv) {
    int i;
    Stack s = new Stack(4);

    for (i = 0; i < 5; i++)
        s.push(i);
}

// What happens at run time?
```

Carnegie Mellon

7

## Catching an Exception

---

```
int main(int argc, char **argv) {
    int i;
    Stack s = new Stack(4);
    try {
        for (i = 0; i < 5; i++) Try statement
            s.push(i);
    }
}
```

Carnegie Mellon

8

## Catching an Exception

---

```
int main(int argc, char **argv) {
    int i;
    Stack s = new Stack(4);
    try {
        for (i = 0; i < 5; i++)
            s.push(i);
    }
    catch(Stack.Full) {
        cerr << " Stack.Push: Stack overflow." << er
        cerr << "Continuing..." << endl;
    }
    Catch Statement
}
```

Carnegie Mellon

9

# Catching Exceptions

---

- After a try statement
  - can have as many catch statements as desired
  - they are checked in order, top to bottom
  - first match found is executed
- Common Practice
  - Collect all executable statements in try statement
  - Deal with all errors in catch statements at the end of the try statement

Carnegie Mellon

10

## Class Declaration -- 2 --

---

```
class Stack {
public:
    class Full{
        const char message[] =
            "Stack.push: Stack overflow.";
    }

    class Empty{
        const char message[] = "...";
    }

    void push(int i) throw(Full);
    int pop() throw(Empty);
private: ....
}
```

Carnegie Mellon

11

## Catching an Exception -- 2 --

---

```
int main(int argc, char **argv) {
    int i;
    Stack s = new Stack(4);
    try {
        for (i = 0; i < 5; i++)
            s.push(i);
    }
    catch (Stack.Full e) {
        cerr << e.message << endl;
        cerr << "Continuing..." << endl;
    }
}
```

Carnegie Mellon

12

## Error Message Policy v. 1

---

- Each exception class contains a field named message which is a character array containing an error message that begins with <Class Name>::<Method Name>:
  - Example: Stack::Push: Stack overflow.
- Method throwing exception does not write message
- If exception signals end of processing, just end, don't write message
- If exceptions signals a real error:
  - write message contained in exception
  - throw another exception which contains an appropriate error message string
- Advantage:
  - catching method decides if a message in exception should be written
  - catching method knows more about what the event means

Carnegie Mellon

13

## Error Message Policy v. 2

---

- Just before throwing exception, a message is written.
- Advantages:
  - Always a trace back of every message that could be written
  - Simple code in object that throws exception
- Disadvantage
  - Might write error message when there was no error

Carnegie Mellon

14