

# CS 240

## Data Structures and Algorithms I

Alex Vondrak

`ajvondrak@csupomona.edu`

October 19, 2011

# Stacks

## Abstract Data Type

```
interface Stack {  
    public void push(int value);  
  
    public int pop()  
        throws StackUnderflowException;  
  
    public int top()  
        throws StackUnderflowException;  
  
    public boolean isEmpty();  
  
    public int size();  
}
```

# Stacks

## Abstract Data Type

```
interface Stack {  
    public void push(int value);  
  
    public int pop()  
        throws StackUnderflowException;  
  
    public int top()  
        throws StackUnderflowException;  
  
    public boolean isEmpty();  
  
    public int size();  
}
```

# Classes

- In Java, classes form a **hierarchy**
- **class** A **extends** B  $\implies A \subseteq B$
- At the top of this hierarchy is Object
- Variables can be **converted** between classes

## Example (Widening Conversion)

```
String s = new String("Something");  
Object obj;  
  
obj = s;
```

# Classes

- In Java, classes form a **hierarchy**
- **class** A **extends** B  $\implies A \subseteq B$
- At the top of this hierarchy is Object
- Variables can be **converted** between classes

## Example (Narrowing Conversion)

```
String s = new String("Something");  
Object obj;  
  
obj = s;  
  
s = new String("Different");  
  
s = obj;
```

# Classes

- In Java, classes form a **hierarchy**
- **class** A **extends** B  $\implies A \subseteq B$
- At the top of this hierarchy is Object
- Variables can be **converted** between classes

## Example (Narrowing Conversion)

```
String s = new String("Something");  
Object obj;  
  
obj = s;  
  
s = new String("Different");  
  
s = (String) obj;
```

# Wrapper Classes

(Almost) Everything's An Object™

- **boolean**
- **byte**
- **char**
- **double**
- **float**
- **int**
- **long**
- **short**

# Wrapper Classes

(Almost) Everything's An Object™

- **boolean** Boolean
- **byte** Byte
- **char** Character
- **double** Double
- **float** Float
- **int** Integer
- **long** Long
- **short** Short



# Wrapper Classes

## Boxing and Unboxing

- Primitive types can be **boxed** into their wrapper classes
- Wrapper objects can be **unboxed** into primitive types

### Example (Manual Boxing/Unboxing)

```
int i = 42;
int j;
Integer k;
k = new Integer(i);
j = k.intValue();
```

# Wrapper Classes

## Boxing and Unboxing

- Primitive types can be **boxed** into their wrapper classes
- Wrapper objects can be **unboxed** into primitive types

### Example (Automatic Boxing/Unboxing)

```
int i = 42;
int j;
Integer k;
k = i;
j = k;
```

## Generalizing Stacks—First Try

```
interface Stack {  
    public void push(Object value);  
  
    public Object pop()  
        throws StackUnderflowException;  
  
    public Object top()  
        throws StackUnderflowException;  
  
    public boolean isEmpty();  
  
    public int size();  
}
```