Exam 1

Solutions posted on the course web page

Today

- Exams
- More objects
- For loops

Short Questions?

Quiz

Exam

- Solution set has been posted on the course web page
- Most common problems:
  - Keeping track of types and type conversions
  - Following the instructions for the coding exercises
- [insert histogram]

Primitive Data Types

Primitive data types are the fundamental units of representation of information (they are the atoms!)

- Operations on primitive data types are implemented directly in hardware (most of the time)
- Fixed size
- Parameter values are copied when calling a method
- Allocated on the stack (private memory area for each method) or the heap (a shared memory area)

Objects

Objects are constructed from primitives and other objects

- Sizes vary
  - Different instances of the same class can be different
  - An instance can change its size at run-time
- References are copied when calling a method, but it is the same object being referred to inside and outside the method
- Allocated on the heap only using a ***constructor***
- A class has its own set of methods associated with it

Immutable vs Mutable Objects

- Immutable objects:
  - The constructor call entirely determines the “value” of the object
  - After construction: cannot be changed – ever!
- Mutable objects:
  - Some methods can change the object after it is constructed

Object Example I

Write a method that returns the past tense of a verb

- Assume regular verbs
- Verbs can end with any letter

Object Example II

Write a method that ***turns*** a verb into the past tense of itself

- Assume regular verbs
- Verbs can end with any letter
- The original object should be modified

While Loops

```
while<LOOP CONDITION>)
{
     <STATEMENTS>
}
```

Loops

- Many of the while loops that we have written involve some index variable that is changed in a consistent way every time we move through the loop
- We would like some consistent way of writing such loops

For Loops

- For loops introduce initialization and update statements.

```
for(<INITIALIZATION>; <LOOP CONDITION>; <UPDATE>)
{
    <STATEMENTS>
}
```

For Loops

- For loops introduce initialization and update statements.

```
          1
for(<INITIALIZATION>; <LOOP CONDITION>; <UPDATE>)
{
    <STATEMENTS>
}
```

For Loops

- For loops introduce initialization and update statements.

```
          1                    2
for(<INITIALIZATION>; <LOOP CONDITION>; <UPDATE>)
{
    <STATEMENTS>
}
```

For Loops

- For loops introduce initialization and update statements.

1 **2:false**

```
for(<INITIALIZATION>; <LOOP CONDITION>; <UPDATE>)
{
    <STATEMENTS>
}
```

3

For Loops

- For loops introduce initialization and update statements.

```
               1                              2:true
for(<INITIALIZATION>; <LOOP CONDITION>; <UPDATE>)
{
        <STATEMENTS>  3
}
```

For Loops

- For loops introduce initialization and update statements.

```
              1                  2:true                 4
for(<INITIALIZATION>; <LOOP CONDITION>; <UPDATE>)
{
      <STATEMENTS>  3
}
```

For Loops

- For loops introduce initialization and update statements.

```
          1                          2:true  5:true      4
for(<INITIALIZATION>; <LOOP CONDITION>; <UPDATE>)
{
      <STATEMENTS>  3
}
```

For Loops

- For loops introduce initialization and update statements.

```
          1                          2:true  5:true      4
for(<INITIALIZATION>; <LOOP CONDITION>; <UPDATE>)
{
     <STATEMENTS>  3  6
}
```

For Loops

- For loops introduce initialization and update statements.

```
          1                          2:true  5:true      4  7
for(<INITIALIZATION>; <LOOP CONDITION>; <UPDATE>)
{
      <STATEMENTS>  3  6
}
```

For Loops

- For loops introduce initialization and update statements.

```
          1                     2:true  5:true      4  7
for(<INITIALIZATION>; <LOOP CONDITION>; <UPDATE>)
{                               8:false
     <STATEMENTS> 3  6
}
```

For Loops

- For loops introduce initialization and update statements.

```
                 1                         2:true  5:true        4  7
for(<INITIALIZATION>; <LOOP CONDITION>; <UPDATE>)
{                                             8:false
      <STATEMENTS>  3  6
}

  9
```

Loop Problem

Write a method that will indicate whether all of the characters in a string are lower case

- What is the method prototype?

Loop Example I

Write a method that will indicate whether all of the characters in a string are lower case

- What is the method prototype?
- What is the method implementation?

Loop Example II

- Quiz question

Summary

- For loops: yet another way to implement a loop
  - No additional functionality beyond while or do-while loops
  - Consistent way of writing certain forms of loops

Wrap Up

Due:

- Project 1: Goes out today
- HW 4: due Wednesday
- HW 5: released on Wednesday

Next time:

- Class methods & generics