

Even More Strings!

An Interesting Article

“India's Identification Scheme:
The Magic Number”

<http://www.economist.com/node/21542763>

Midterm Logistics

- Midterm next **Monday, February 13** from **7-9PM**.
 - Last name A - O: Cemex Auditorium
 - Last name P - Z: Braun Auditorium
- Covers material up through and including Friday's lecture.
- Open-book, open-note, closed computer.
- Alternate exam times:
 - Monday, February 13 from 9 - 11AM
 - Tuesday, February 14 from 9 - 11AM.
- Please contact us ASAP if you want to take the exam at an alternate time.

Review Session

- Practice midterm available now.
- Review session: Sunday, February 12 from 7:00PM – 8:30PM in Hewlett 200.

|oghnamōrλuvzīgriym|érzueenjyra=rzrphmhmzy|úλx
 zufrzrziπhλurhχ|nōm̄yχlδa+b+uīuāst~~α~~λx+wripen|nripr̄r̄
 ππi|nxiδuλh̄ȳczπh̄jδōλh̄j̄nū|ōph̄r̄am̄ōzλu|h̄|ōt̄h̄n̄o=rzκ
 hē|zūn:sc̄φ|λfπ̄īz̄t. Anzuvōim̄z̄ȳπriλurzh̄|w. λ|φ̄īδ̄c̄r̄p̄
 c̄p̄duoz̄r̄|nū|sg|p̄ȳgōp̄dēt̄w̄δ̄c̄h̄r̄h̄δ̄īλb̄ȳ+r̄c̄|p̄|u|f̄x̄n̄z̄ḡt̄z̄h̄p̄u|h̄
 =r̄|j̄īs̄b̄h̄|f̄:zē|l̄s̄ū̄ȳx̄uez̄r̄ig:z̄c̄ū̄λ̄ḡm̄īz̄r̄r̄=+r̄p̄ō̄z̄x̄h̄ḡp̄ȳ
 πriλf̄z̄h̄āgr̄ē̄h̄ā̄st̄h̄r̄z̄h̄ī=|s̄z̄ut. Δ̄īj̄n̄īλ̄x̄ḡz̄n̄|d̄j̄h̄t̄h̄oē
 • p̄ū̄ōḡz̄īō|ū̄+c̄n̄x̄x̄κ̄m̄īc̄:κ̄ȳr̄ō+|r̄p̄h̄r̄z̄λ̄s̄λ̄c
 • x̄n̄z̄f̄s̄=h̄p̄m̄h̄d̄ūd̄s̄ā̄īλ̄ūb̄+z̄h̄r̄p̄z̄ḡs̄x̄d̄=r̄λ̄ū̄|f̄īz̄ō̄ḡūb̄h̄t̄c̄u
 • κ̄n̄m̄x̄īc̄d̄r̄λ̄ūv. Δ̄l̄c̄īn̄h̄λ̄z̄ū̄z̄h̄n̄:ō̄j̄m̄e.
 z̄p̄r̄v̄π̄īz̄ȳd̄īz̄īr̄z̄ā. λ̄b̄c̄p̄|λ̄f̄h̄h̄+d̄|ōv̄z̄h̄n̄:w̄p̄ūr̄x̄|f̄īc̄:u
 c̄=ḡp̄b̄c̄īh̄j̄c̄īr̄x̄|n̄f̄+)=d̄f̄r̄īp̄o=r̄z̄b̄ḡh̄r̄z̄ū̄+λ̄p̄ē̄r̄n̄z̄j̄ū̄+ōn̄īu. Δ̄
 t̄z̄ē̄ḡī+t̄r̄ī=|m̄~~z̄~~z̄īn̄|κ̄r̄x̄ā̄b̄. Δ̄ō̄j̄n̄īλ̄c̄x̄m̄z̄f̄π̄ūḡ. λ̄h̄=
 λ̄h̄r̄ōd̄īp̄ȳ|h̄ō̄λ̄āt̄+q̄d̄ē̄λ̄s̄z̄ā̄z̄f̄p̄ȳj̄h̄d̄īz̄ūz̄ō. λ̄l̄z̄ūp̄h̄ḡz̄īπ̄r̄ī
 λ̄ūv̄z̄n̄+η̄λ̄f̄|δ̄c̄:ī̄r̄:λ̄ō̄z̄z̄h̄=|ū+λ̄κ̄z̄ūb̄m̄ō̄r̄λ̄ūf̄:
 • π̄p̄z̄ū̄j̄c̄ī=λ̄m̄ā̄l̄ū̄n̄p̄r̄x̄z̄z̄b̄|δ̄c̄:n̄z̄h̄ē̄|ō̄j̄m̄s̄p̄ȳr̄ūd̄ḡh̄ō̄
 • +h̄īc̄|ḡh̄ḡz̄īz̄|m̄p̄īc̄|c̄s̄=ḡē̄h̄z̄ūh̄p̄ō̄j̄f̄c̄p̄r̄c̄īr̄ūw. Δ̄ōō=m̄z̄h̄ȳm̄l̄
 λ̄h̄r̄

ε̄p̄īλ̄
h̄c̄r̄

• λ̄h̄r̄ā̄z̄p̄r̄ī=|r̄ȳr̄λ̄h̄ōūc. Δ̄f̄|h̄r̄r̄x̄īp̄z̄h̄ī=r̄ūēōh̄ū̄j̄ō̄f̄ō̄ūb̄d̄ō̄r̄īh̄
 • p̄īz̄λ̄h̄|s̄īōm̄īπ̄r̄īr̄d̄ō̄ḡl̄=|m̄z̄l̄h̄īḡūīh̄r̄t̄z̄p̄z̄ā̄z̄x̄j̄ō̄h̄λ̄f̄īl̄:|p̄ḡē̄z̄t̄p̄r̄ī
 • z̄v̄j̄n̄d̄ō̄r̄ḡū̄īḡūp̄m̄h̄īr̄ūd. λ̄x̄p̄ā̄z̄ḡc̄p̄r̄c̄h̄r̄ūc. Δ̄z̄x̄r̄d̄ū̄r̄:u
 • b̄+q̄|:ē̄c.
 V̄z̄ā̄z̄x̄t̄r̄īl̄:m̄īh̄p̄ū̄c̄ā̄h̄z̄ē̄λ̄ḡīh̄m̄d̄īc̄|z̄ō̄n̄:p̄ȳr̄ḡz̄ȳḡūf̄d̄p̄m̄:
 s̄ā̄īr̄ūf̄z̄īr̄w. λ̄c̄=|m̄π̄v̄z̄z̄. λ̄āx̄h̄r̄p̄λ̄h̄r̄h̄t̄+r̄z̄h̄ī|f̄īn̄r̄h̄z̄ē̄n̄. Δ̄ī
 n̄īā̄c̄r̄ū̄|b̄ȳ|r̄p̄h̄īn̄h̄ȳ|s̄ō̄z̄īū̄m̄d̄āḡh̄z̄ū̄+ḡz̄|d̄r̄x̄r̄d̄s̄z̄ā̄j̄g. Δ̄ōō
 p̄r̄d̄c̄q̄j̄ē̄l̄b̄z̄h̄īκ̄|q̄r̄ḡh̄t̄+|s̄ā̄l̄īz̄|ḡx̄ḡ|r̄r̄ūd̄x̄ā̄h̄ū̄ȳ+ūp̄r̄āz̄|s̄īλ̄=|ȳ
 ō̄r̄ūām̄īō̄r̄h̄īz̄r̄s̄p̄ūḡm̄īn̄h̄z̄ūb̄d̄ā̄j̄|p̄ḡz̄f̄p̄λ̄n̄|h̄ē̄x̄ū̄c̄|āz̄n̄z̄p̄d̄h̄z̄
 p̄λ̄h̄h̄t̄+r̄z̄h̄ā̄x̄p̄r̄c̄īx̄p̄ȳ|ūē̄īm̄ḡ=ḡp̄n̄v̄r̄z̄īz̄|ḡs̄=|b̄l̄ō̄h̄z̄p̄r̄ḡn̄h̄r̄
 z̄īp̄l̄ē̄n̄t̄īl̄:s̄ū̄f̄ō̄|r̄b̄īz̄=|s̄x̄s̄ȳr̄ḡūīm̄z̄m̄t̄|λ̄p̄h̄t̄+|d̄n̄ā̄p̄|ḡh̄c̄
 c̄ūd̄īr̄m̄ū̄|ūz̄ūc̄x̄c̄q̄d̄īx̄r̄p̄z̄z̄ō̄j̄f̄z̄īz̄h̄d̄ȳē̄j̄ūz̄ā. λ̄n̄p̄r̄īt̄c̄ȳf̄
 λ̄n̄z̄r̄ī+η̄λ̄ḡz̄īz̄c̄Δ̄m̄.
 P̄r̄īm̄ā̄ōōλ̄īz̄m̄λ̄h̄=ē̄b̄
 d̄ē̄r̄:s̄ȳr̄ūd̄r̄ū̄īn̄|v̄. λ̄.
 H̄ō̄īḡḡ-λ̄b̄m̄īz̄z̄f̄m̄η̄r̄λ̄c̄|d̄m̄īō̄h̄c̄p̄h̄t̄īz̄=|r̄c̄ū̄r̄d̄p̄r̄:λ̄ō
 z̄h̄īp̄r̄īḡz̄n̄|z̄ē̄d̄t̄ȳf̄:|ā̄b̄|s̄λ̄īd̄m̄c̄|p̄ūḡx̄r̄z̄x̄t̄r̄ē̄h̄p̄ūc̄d̄an̄:ā̄ī|ō̄ḡ

Encryption

The Caesar Cipher

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Key: +2

The Caesar Cipher

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B

Key: +2

H	E	L	L	O
J	G	N	N	Q

The Caesar Cipher

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B

Key: -2

J G N N Q

The Caesar Cipher

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X

Key: -2

J	G	N	N	Q
H	E	L	L	O

The Caesar Cipher

- Encryption key is a **shift number**.
- Cycle all letters forward by the shift number to encrypt.
- Cycle all letters backward by the shift number to decrypt.

Limits of the Caesar Cipher

- Every letter is always replaced by the same replacement letter.
- Using the clever technique of **frequency analysis**, it is possible to guess the key.
 - The most frequent letter is probably going to be E.
 - Knowing what E maps to, we can find the key by looking at the shift of E.
- ابو يوسف يعقوب بن اسحاق الصبّاح الكندي (abu Yusuf Ya'qūb ibn Ishāq al-Sabbah al-Kindī) determined how to do this in the 9th century.

Breaking the Caesar Cipher

DL JOVVZL AV NV AV AOL TVVU PU AOPZ KLJHKL
HUK KV AOL VAOLY AOPUNZ, UVA ILJHBZL AOL
HYL LHZF, IBA ILJHBZL AOL HYL OHYK,
ILJHBZL AOHA NVHS DPSS ZLYCL AV VYNHUPGL
HUK TLHZBYL AOL ILZA VM VBY LULYNPLZ HUK
ZRPSSZ, ILJHBZL AOHA JOHSSLUNL PZ VUL AOHA
DL HYL DPSSPUN AV HJLWA, VUL DL HYL
BUDPSSPUN AV WVZAVVUL, HUK VUL DOPJO DL
PUALUK AV DPU, HUK AOL VAOLYZ, AVV. (QMR)

Useful `String` Methods

`int length()`

Returns the length of the string

`char charAt(int index)`

Returns the character at the specified index. Note: Strings indexed starting at 0.

`String substring(int p1, int p2)`

Returns the substring beginning at **`p1`** and extending up to but not including **`p2`**

`String substring(int p1)`

Returns substring beginning at **`p1`** and extending through end of string.

`boolean equals(String s2)`

Returns true if string **`s2`** is equal to the receiver string. This is case sensitive.

`int compareTo(String s2)`

Returns integer whose sign indicates how strings compare in lexicographic order

`int indexOf(char ch) or int indexOf(String s)`

Returns index of first occurrence of the character or the string, or -1 if not found

`String toLowerCase() or String toUpperCase()`

Returns a lowercase or uppercase version of the receiver string

So What's With the Suit?

Class

class

Objects Revisited

- An object is a combination of
 - **State** – persistent information, and
 - **Behavior** – the ability to operate on that state.
- **GRect** state:
 - Position
 - Size
 - Color
 - Is filled?
 - etc.
- **GRect** behavior:
 - Move
 - Change color
 - Change fill state
 - Tell position
 - etc.

Objects Revisited

- An object is a combination of
 - **State** – persistent information, and
 - **Behavior** – the ability to operate on that state.
 - **GPoint** state:
 - Position
 - **GPoint** behavior:
 - Move
 - Move by angle
 - Tell x
 - Tell y

Objects Revisited

- An object is a combination of
 - **State** – persistent information, and
 - **Behavior** – the ability to operate on that state.
- **String** state:
 - Character sequence
- **String** behavior:
 - Get characters
 - Produce substring
 - etc.

Classes and Objects

- Each object is an **instance** of some **class**.
- The class determines
 - what state each instance maintains.
 - what behaviors each instance possesses.
- Each instance determines
 - the specific values for that state information.

Creating our own Class



Creating our own Class

- State:

- The current number.

- Behavior:

- Read the counter.
- Increment the counter.

We use instance variables to keep track of state.

Creating our own Class

- State:
 - The current number.

We use instance variables to keep track of state.

- Behavior:
 - Read the counter.
 - Increment the counter.

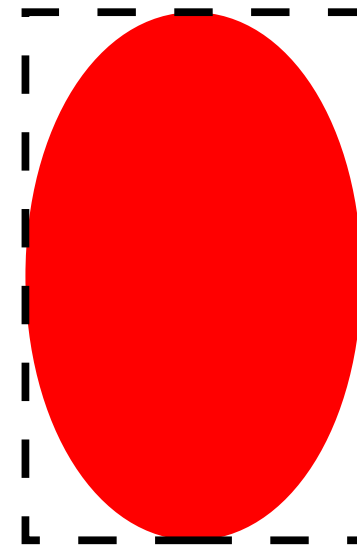
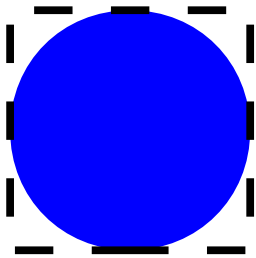
We use methods to specify behavior.

Instance Variables Revisited

- Each instance of a class gets its own, unique copy of each instance variable.
- Different instances of the same object cannot read or write each others' instance variables.

Instance Variables Revisited

- Each instance of a class gets its own, unique copy of each instance variable.
- Different instances of the same object cannot read or write each others' instance variables.



Constructors

- A **constructor** is a special method defined in a class that is responsible for setting up class's instance variables to appropriate values.
- Syntax:

```
public NameOfClass (parameters) {  
    /* ... body of constructor ... */  
}
```

- Inside a constructor:
 - Give initial values to instance variables.
 - Set up instance variables based on values specified in the parameters.
- Constructor called when instance created with **new**.

toString()

- To get a string representation of an object, Java uses a method

```
public String toString()
```

- If you define this method in your Java classes, you can customize what string will be produced.
- Otherwise, you get Icky Javaspeak string representations.