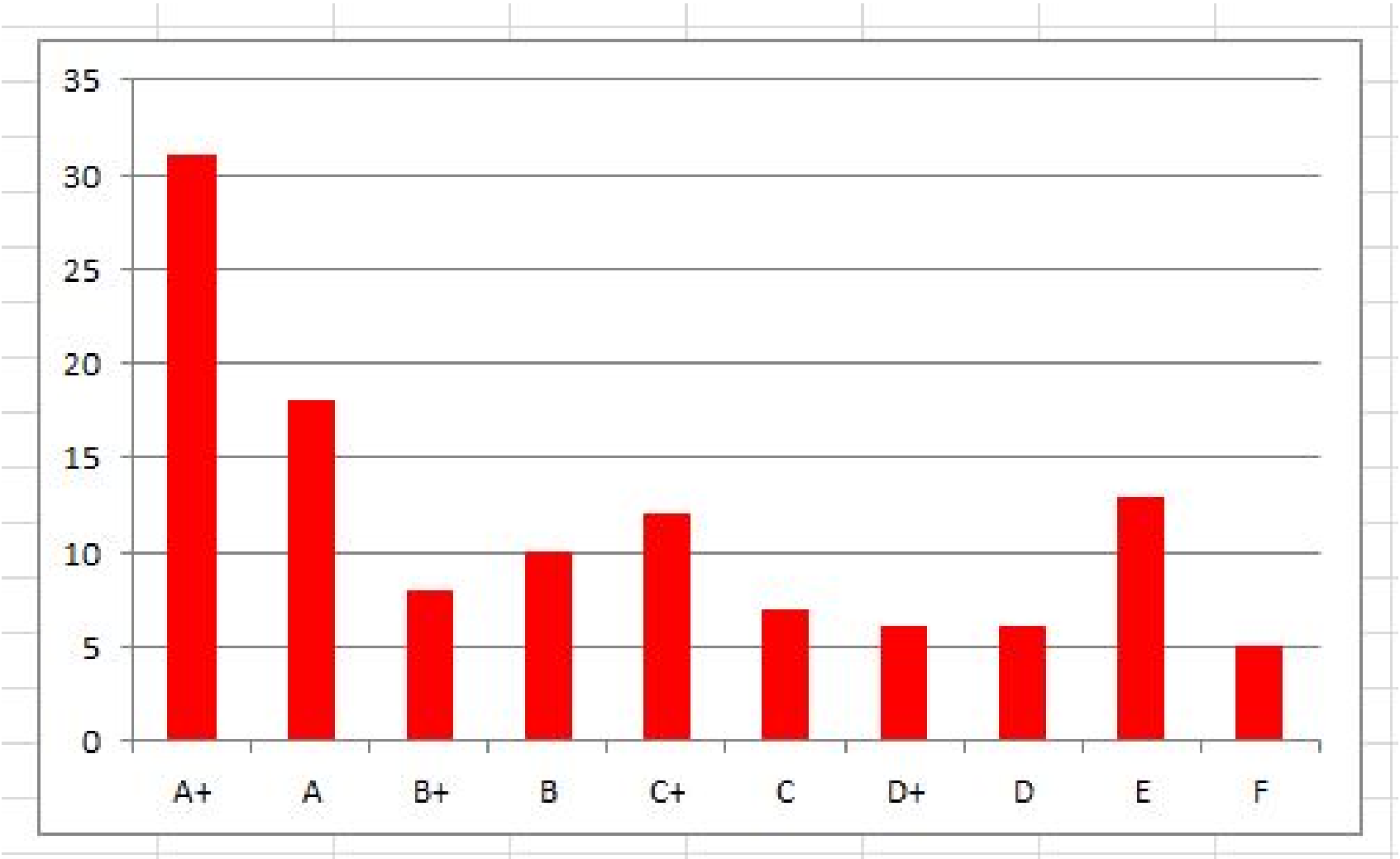
Programming for Mobile Computing

EECS 1022

`moodle.yorku.ca`

Grade distribution of written part of midterm



Anomalous grade distribution

If more than 30% of the students receive an A+ or A, then the grade distribution is deemed anomalous. As a result, expect it to be more difficult to receive an A+ or A in the written part of the final exam.

Mistake in the marking of your midterm

If you believe that there is a mistake made in the marking of your midterm (the marking scheme is available on Moodle), then email the instructor within one week (that is, before Monday July 24). In the email, clearly describe the mistake in marking. Your whole midterm will be reviewed. As a result your mark may go up, stay the same, or go down.

Strings

Strings are immutable objects.

The state of an immutable object cannot be changed.

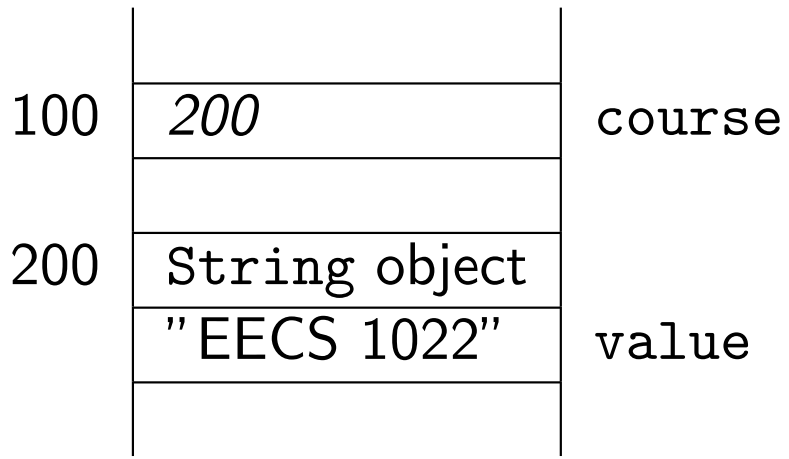
The `String` API does not contain any mutators.

The `StringBuffer` class provides mutable strings. [1]

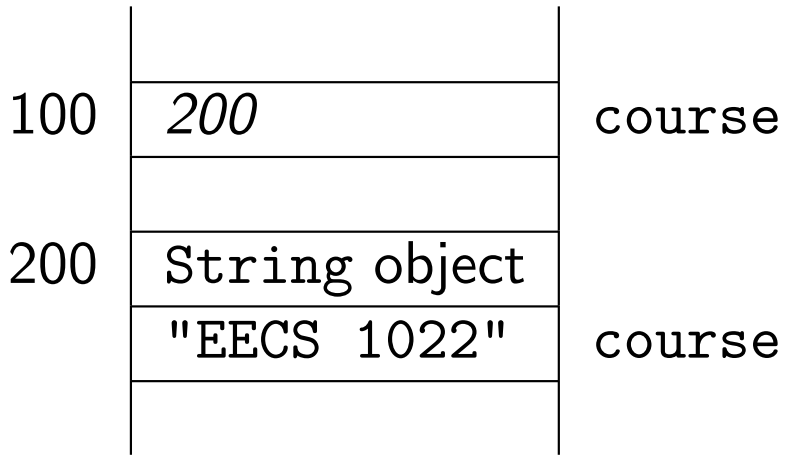
[1] We will come back to the `StringBuffer` class later.

Strings

String course = new String("EECS 1022");



Strings



String reference: `course`
String object: object at address 200
String literal: `"EECS 1022"`

Strings are everywhere

Instead of

```
String course = new String("EECS 1022");
```

we are allowed to write

```
String course = "EECS 1022";
```

Although in most cases you may think of `"EECS 1022"` and `new String("EECS 1022")` as synonyms, they are not always equivalent.[2]

[2] Hardly ever will this difference impact your app.

Strings are immutable

According to the Java Language Specification,

> *Strings that are the values of constant expressions are "interned" so as to share unique instances*

James Gosling, Bill Joy, Guy L. Steele Jr. and Gilad Bracha. The Java Language Specification. Third edition. Addison-Wesley. 2005.

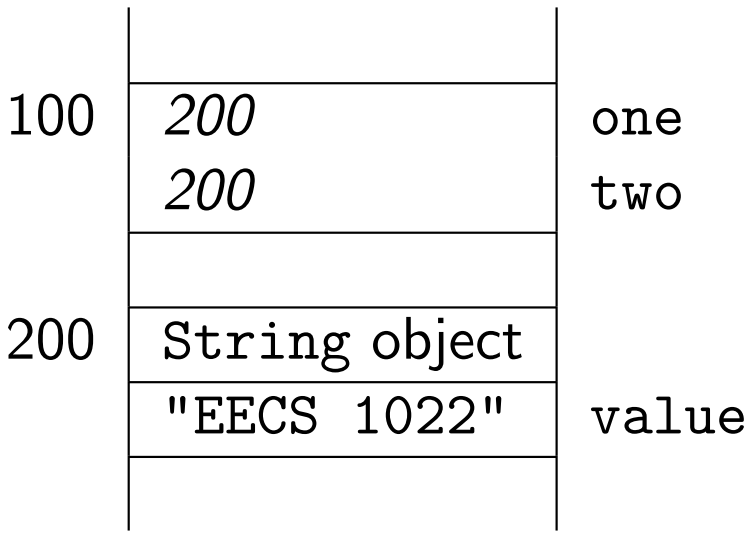
Strings are immutable

> *Strings that are the values of constant expressions are "interned" so as to share unique instances*

These constant expressions are built from String literals and the binary operator $+$.

Strings are immutable

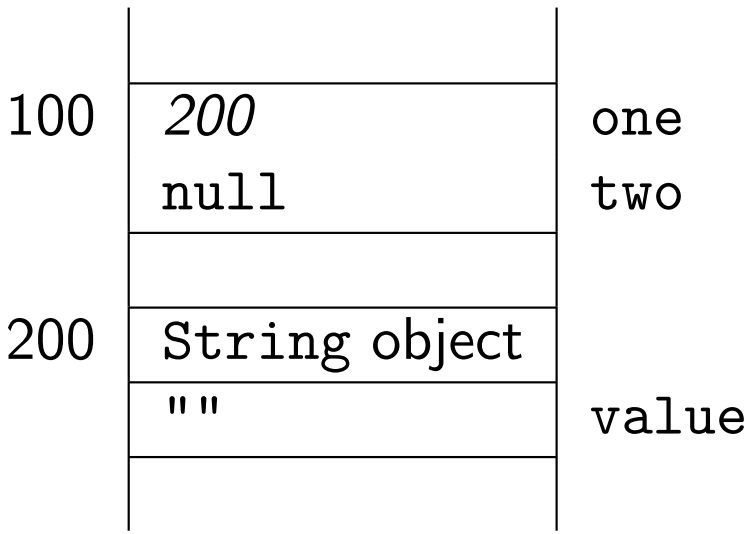
String one = "EECS 1022";
String two = "EECS" + " " + "1022";



This saves memory. Why can `one` and `two` refer to the same `String` object?

The empty string versus `null`

String one = "";
String two = null;



String API

https://docs.oracle.com/javase/8/docs/api/java/lang/String.html

If only I had known . . .

Problem

Print, for example,

```
If I had bought ibm shares on 01/15/68
and sold them on 01/16/93,
I would have made a 1599.88% loss
```

where `ibm` is provided as a command line argument.

Let's start with something simpler

Before we attempt to address that problem, we first solve an array of simpler problems that involve strings.

length method

public int length()

Returns the length of this string.

charAt method

public char charAt(int index)

Returns the char value at the specified index.

charAt method

public char charAt(int index)

Returns the char value at the specified index.

Question

How do you extract the first character of a string?

```
public static char firstChar(String word)
{
    ...
}
```

charAt method

Answer

```
char letter = word.charAt(0);
return letter;
```

charAt method

Answer

```
char letter = word.charAt(0);
return letter;
```

Question

What happens when the string is empty?

charAt method

Answer

```
char letter = word.charAt(0);
return letter;
```

Question

What happens when the string is empty?

Question

What happens when the string is null?

charAt method

public char charAt(int index)

Returns the char value at the specified index.

charAt method

public char charAt(int index)

Returns the char value at the specified index.

Question

How do you extract the last character of a string?

```
public static char lastChar(String word)
{
    ...
}
```

charAt method

Answer

```
char letter = word.charAt(word.length() - 1);
return letter;
```

charAt method

Answer

```
char letter = word.charAt(word.length() - 1);
return letter;
```

Question

What happens when the string is empty?

charAt method

Answer

```
char letter = word.charAt(word.length() - 1);
return letter;
```

Question

What happens when the string is empty?

Question

What happens when the string is null?

String Problems

Question

How do you count the number of a's in a string?

```
public static int numberOfAs(String word)
{
    ...
}
```

Answer

```
count = 0;
for (int i = 0; i < word.length(); i++)
{
   if (word.charAt(i) == 'a')
   {
      count++;
   }
   return count;
}
```

String Problems

Question

How do you count the number of a given letter in a string?

```
public static int numberOf(String word, char letter)
{
    ...
}
```

Answer

```
count = 0;
for (int i = 0; i < word.length(); i++)
{
   if (word.charAt(i) == letter)
   {
      count++;
   }
   return count;
}
```

String Problems

Question

How to check if a string contains only digits?

```
public static boolean containsOnlyDigits(String word)
{
    ...
}
```

Answer

```
boolean onlyDigits = true;
for (int i = 0; i < word.length(); i++)
{
  char letter = word.charAt(i);
  if (letter < '0' || letter > '9')
  {
    onlyDigits = false;
  }
}
return onlyDigits;
```

Answer

```
boolean onlyDigits = true;
for (int i = 0; i < word.length() && onlyDigits; i++)
{
  char letter = word.charAt(i);
  if (letter < '0' || letter > '9')
  {
    onlyDigits = false;
  }
}
return onlyDigits;
```

Answer

```
boolean onlyDigits = true;
for (int i = 0; i < word.length() && onlyDigits; i++)
{
   char letter = word.charAt(i);
   if (!Character.isDigit(letter))
   {
      onlyDigits = false;
   }
}
return onlyDigits;
```

String Problems

Answer

```
boolean onlyDigits = true;
for (int i = 0; i < word.length() && onlyDigits; i++)
{
  char letter = word.charAt(i);
  onlyDigits = Character.isDigit(letter);
}
return onlyDigits;
```

String Problems

Question

How to remove a given character from a string?

```
public static String remove(String word, char letter)
{
    ...
}
```

Answer

```
String result = "";
for (int i = 0; i < word.length(); i++)
{
   char other = word.charAt(i);
   if (other != letter)
   {
      result = result + other;
   }
}
```

String Problems

Question

How to reverse a string?

```
public static String reverse(String word)
{
    ...
}
```

String Problems

Answer

```
String reversal = "";
for (int i = 0; i < word.length(); i++)
{
   char letter = word.charAt(i);
   reversal = letter + reversal;
}
return reversal;
```

String Problems

Question

How to check if a string has duplicate characters?

```
public static boolean hasDuplicates(String word)
{
    ...
}
```

Answer

```
boolean found = false;
for (int i = 0; i < word.length(); i++)
{
  char one = word.charAt(i);
  for (int j = 0; j < word.length(); j++)
  {
    char other = word.charAt(j);
    if (i != j && one == other)
    {
      found = true;
    }
  }
}
return found;
```

Answer

```
boolean found = false;
for (int i = 0; i < word.length(); i++)
{
  char one = word.charAt(i);
  for (int j = i + 1; j < word.length(); j++)
  {
    char other = word.charAt(j);
    if (one == other)
    {
      found = true;
    }
  }
}
return found;
```

String Problems

Answer

```
boolean found = false;
for (int i = 0; i < word.length() && !found; i++)
{
  char one = word.charAt(i);
  for (int j = i + 1; j < word.length() && !found; j++)
  {
    char other = word.charAt(j);
    if (one == other)
    {
      found = true;
    }
  }
}
return found;
```

Answer

```
boolean found = false;
for (int i = 0; i < word.length() && !found; i++)
{
  char one = word.charAt(i);
  for (int j = i + 1; j < word.length() && !found; j++)
  {
    char other = word.charAt(j);
    found = one == other;
  }
}
return found;
```

String Problems

Question

How to print duplicate characters from a string?

```
public static void duplicates(String word)
{
    ...
}
```

For example `duplicates("aaabccdee")` prints

```
aaaccee
```

String Problems

Answer

```
for (int i = 0; i < word.length(); i++)
{
  char one = word.charAt(i);
  boolean found = false;
  for (int j = 0; j < word.length(); j++)
  {
    char other = word.charAt(j);
    if (i != j && one == other)
    {
      found = true;
    }
  }
  if (found)
  {
    System.out.print(one);
  }
```

String Problems

Answer

```
for (int i = 0; i < word.length(); i++)
{
  char one = word.charAt(i);
  boolean found = false;
  for (int j = 0; j < word.length() && !found; j++)
  {
    char other = word.charAt(j);
    if (i != j && one == other)
    {
      found = true;
      System.out.print(one);
    }
  }
}
```

String Problems

Question

How to print duplicate characters from a string?

```
public static void duplicates(String word)
{
    ...
}
```

For example `duplicates("aaabccdee")` prints

```
ace
```

Answer

```
for (int i = 0; i < word.length(); i++)
{
  char one = word.charAt(i);
  boolean alreadyPrinted = false;
  for (int j = 0; j < i && !alreadyPrinted; j++)
  {
    char other = word.charAt(j);
    alreadyPrinted = one == other;
  }
```

Answer

```
  boolean duplicateFound = false;
  for (int j = i + 1;
       j < word.length && !alreadyPrinted && !duplicateFound;
       j++)
  {
    duplicateFound = one == other;
  }
  if (!alreadyPrinted && duplicateFound)
  {
    System.out.println(one);
  }
}
```

String Problems

Question

How to convert a numeric string to an int?

```
public static int toInt(String word)
{
    ...
}
```

Answer

```
int result = 0;
int factor = 1;
for (int i = word.length() - 1; i >= 0; i--)
{
  char letter = word.charAt(i);
  int digit = letter - '0';
  result = result + factor * digit;
  factor = factor * 10;
}
return result;
```