The following code is what type of method?

```java
public Student(String theName) {
    this.name = theName;
}
```

A. Accessor
B. Constructor
C. Modifier
D. Getter
E. Initializer
CREATING CLASSES

CREATING A COURSE CLASS

• An instructor has students in each class course.
  – Each course can have different students.
• For a course we might want to know
  – The instructor name
  – The course number
  – The students in that period

UML CLASS DIAGRAM

• There is a standard way to diagram object-oriented classes
  – It is a UML class diagram
    • It shows the classes as boxes and the relationships between them

  Shows inheritance
  “Has a” or association

CREATING A CLASSPERIOD CLASS

• We want fields for the
  – Instructor’s name
  – Students in the period
• What type should each of these be?
  – A name can be a string
  – The students in the period can be an array of Student objects.
    • With a max of 35 students in a class period.
CREATE A CLASS EXERCISE

• Declare a new class Course in the file Course.java
• Add the fields:
  – private String instructor;
  – private Student[] students = new Student[35];

CREATE A CLASS EXERCISE - CONTINUED

• Add constructors to ClassPeriod
  – Add the no argument constructor
  – Add a constructor that takes the teacher’s name and the period number
• Remember that constructors are declared as
  public ClassName(paramList)
  {
  
  }

ADD ACCESSORS AND MODIFIERS

• Add methods to get and set the fields in the class period
  public String getInstructor()
  {
    return this.instructor;
  }

  public void setInstructor(String instructor)
  {
    this.instructor = instructor;
  }

OVERRITE TOSTRING

• Add the following method to Course
  public String toString()
  {
    
  }

  Add code to the method to return
  – a String with the instructor
  – the course number
  – the number of students in the course
• Also for the method:
  – The length of the array isn’t a count of the actual students
  – Find out how many in the array are not null
SUMMARY

- Object-oriented programs
  - Have interacting objects
- Classes define and create objects
  - The contain the fields, constructors, and methods
    - Some methods are called accessor and modifier methods since they give you access to private fields and let you modify private fields
- A class inherits methods from a parent class

TEXT AS UNIMEDIA

- Computers only understand 0 and 1
  - On and off, using magnetism, voltage, etc.
- But we can represent anything with those:
  - Text, Pictures, Sounds, Movies…

JAVA.LANG.STRING

- Text in Java is stored as a String object
  - In a format called Unicode
    - It uses 2 bytes per character
- HTML is a text-based language.
  - It is used to show pictures and play sounds.
- We can do the same with text:
  - Convert a picture to text.
  - Convert a sound to text.
A string literal is enclosed in double quotes

```java
String message = "Hi There";
```

How do you add a double quote to a string?

- for example to express:  
  ```java
  She said, "Hi there"
  ```

To do this, use `\"`


```java
    String s = "She said, \"Hi there\"";
    System.out.println(s);
    She said, "Hi there"
```

Other special characters:

- `\n` for new line
- `\t` for tab

Strings are sequences of characters

- You can get the character at an index in the string
  - Starting with index 0
    ```java
    stringObj.charAt(index);
    ```

Try this:

```java
    String test = "Hello";
    System.out.println(test.charAt(0));
    System.out.println(test.charAt(4));
```

- How would you get the second character?

API

- An API is an Application Programming Interface.

- Basically, it lists and describes all of the methods and classes available in that language.

- Open the Java API
  ```
  http://java.sun.com/j2se/1.5.0/docs/api/
  ```

  - Click on the java.lang package
  - Click on the String class
WORKING WITH DELIMITED STRINGS

• Sometimes you get information about an object
  – In the form of a delimited string:

  Jane Dorda:88,92,95,87,93,85
  Mike Koziatek:75,92,83,81,91,87
  Sharquita Edwards:91,93,95,92,94,99

• Here the delimiters are
  – a colon after the name, and
  – commas between the grades.

PARSING A DELIMITED STRING

• Add another constructor to the Student class
  – That takes a delimited string of the form
    
    Name : grade1, grade2, grade3, grade4, grade5
  

STRING METHODS

• Look at the methods available for the String class.
  – Which will return part of a string?
  – Which will return the first index of a list of characters in the string?
  – While will return the last index of a list of characters?
  – Which will remove extra space before and after any other characters?
  – Which will return an array of String objects -
    • By chopping the string up into substrings!
    • Based on specified delimiters (like spaces or commas)!

PARSING A DELIMITED STRING

• Use the split() method to get an array of Strings
  – First based on the colon delimiter
• Use trim() to clear off any additional space from the name
  – The name is the first element in the returned array
• Use the split() method again to get the array of grades as strings
  – The grades will be the second element of the returned array
  – Use the comma as the delimiter
• Use Double.parseDouble() to translate the grade string into a double value
  – For the grade array
CONVERTING TO A NUMBER

- Strings are stored in Unicode format
  - Two bytes per character
- Integers are stored in 4 bytes (32 bits)
- You need to convert a number that is represented as a string into the number representation.
- The wrapper classes have methods to do this
  - `Integer.parseInt(numStr)`

The string "1234" is stored in 8 bytes
With each character taking 2 bytes

bytes ——> 0|1|0|2|0|3|0|4

The integer 1234 is stored in 4 bytes

bytes ——> 0|0|4|214

Where the 4 is really
4 * 256 = 1024
So,
1024 + 214 = 1234

public Student(String delim)
{
    // split string based on name delimiter
    String[] split1 = delim.split(";");
    this.name = split1[0].trim();

    // get the grade string and break it and convert to double
    String[] split2 = splitArray[1].split(",");
    this.grades = new double[split2.length];
    for(int i = 0; i < split2.length; i++)
        this.grades[i] = Double.parseDouble(split2[i]);
}

TESTING THE CONSTRUCTOR

- Write a main method that will create a Student object and initialize the name and grade array
  - From a delimited string
- Run the main method from DrJava
- Use the Debugger to walk through the constructor

ICLICKER QUESTION
What will be the output of the following Java code?

```java
String quizString = "I wonder how \nthis \nwill "look"?";
System.out.println(quizString);
```

A. I wonder how this will "look"?

B. I wonder how nthis nwill "look"?

C. "I wonder how this will "look"?"

D. I wonder how this will "look"?

E. I wonder how this will look?