

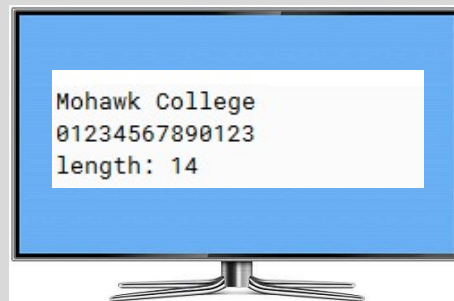


String Static Methods - length()

```
public static void main (String[] args)
{ //main
  String phrase = "Mohawk College";

  //index numbers
  System.out.println( phrase );
  System.out.println("01234567890123" );

  System.out.println(" length: " + phrase.length() );
} //main
```

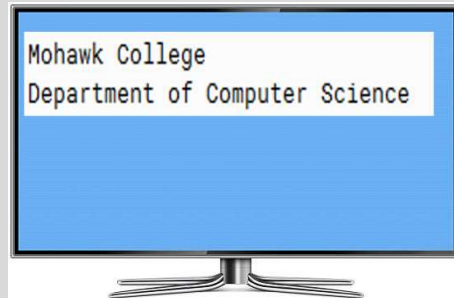


String Static Methods - the + sign

```
// + sign means concatenation
String phrase = "Mohawk College";

System.out.println(phrase + "\n" + "Department of Computer Science");

}
```

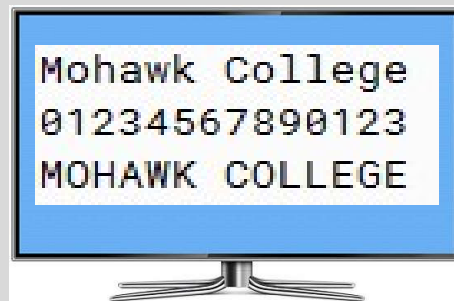


String Static Methods - toUpperCase()

```
public static void main (String[] args)
{ //main
  String phrase = "Mohawk College";

  //index numbers
  System.out.println( phrase );
  System.out.println("01234567890123" );

  //upper case
  System.out.println(phrase.toUpperCase() );
} //main
```



String Static Methods - contains()

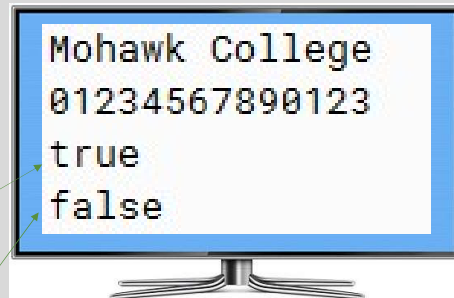
```
public static void main (String[] args)
{ //main
    String phrase = "Mohawk College";

    //index numbers
    System.out.println( phrase );
    System.out.println("01234567890123" );

    //contains
    System.out.println(phrase.contains("w" ) );

    System.out.println(phrase.contains("Academy" ) );

} //main
```



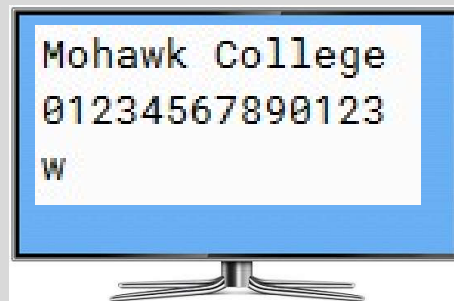
String Static Methods - charAt()

```
public static void main (String[] args)
{ //main
    String phrase = "Mohawk College";

    //index numbers
    System.out.println( phrase );
    System.out.println("01234567890123" );

    //character index number
    System.out.println( phrase.charAt(4) );

} //main
```



String Static Methods - indexOf()

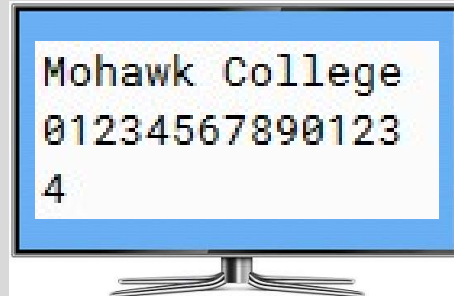
```
public static void main (String[] args)
{ //main

    String phrase = "Mohawk College";

    //index numbers
    System.out.println( phrase );
    System.out.println("01234567890123" );

    //index number
    System.out.println( phrase.indexOf("w") );

} //main
```



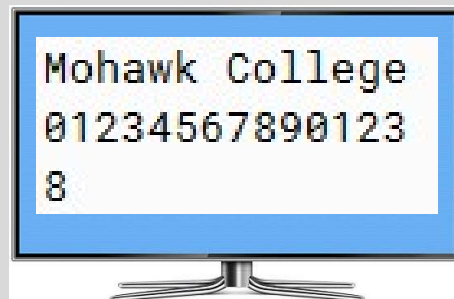
String Static Methods - lastIndexOf()

```
public static void main (String[] args)
{ //main
    String phrase = "Mohawk College";

    //index numbers
    System.out.println( phrase );
    System.out.println("01234567890123" );

    //last index number
    System.out.println( phrase.lastIndexOf("o") );

} //main
```



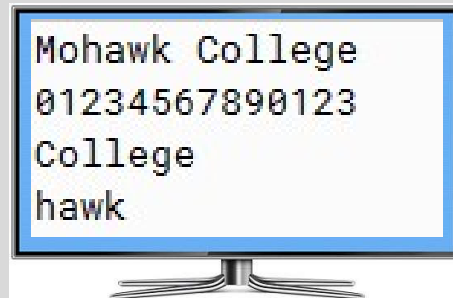
String Static Methods - substring()

```
public static void main (String[] args)
{ //main
  String phrase = "Mohawk College";

  //index numbers
  System.out.println( phrase );
  System.out.println("01234567890123" );

  //substring
  System.out.println( phrase.substring(7) );
  System.out.println( phrase.substring(2,6) );

} //main
```



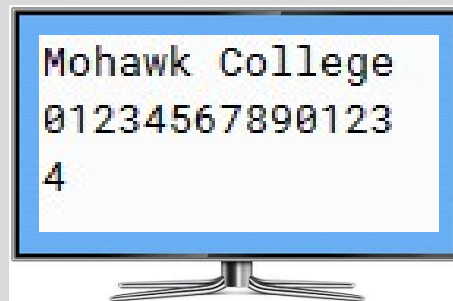
String Static Methods - Combining Methods

```
public static void main (String[] args)
{ //main
  String phrase = "Mohawk College";

  //index numbers
  System.out.println( phrase );
  System.out.println("01234567890123" );

  //combining methods
  System.out.println(phrase.substring(2,6).length() );

} //main
```



Getting User Input into a Variable

```
import java.util.Scanner;

/**
 * This program asks the user to enter their name and age
 *
 * @author James Bond
 * @version v100
 */
public class AskForAge
{ //class

    public static void main (String[] args)
    { //main
        Scanner keybd = new Scanner(System.in);

        String name;
        int age;

        System.out.print("Enter your first name: ");
        name = keybd.nextLine();

        System.out.print("Enter your age " + name + ": ");
        age = keybd.nextInt();
        keybd.nextLine(); //eats up the rest of the line
    } //main
} //class
```

1. Attach the Scanner library
2. Create keybd object of type Scanner
3. Read a string then an integer.



USER Input – it's wise to read in initially as a String and then convert .., -- to prevent program crashes

```
import java.util.Scanner;

/**
 * This program asks the user to enter their name and age
 *
 * @author James Bond
 * @version v100
 */
public class AskForAge
{ //class

    public static void main (String[] args)
    { //main
        Scanner keybd = new Scanner(System.in);

        String name;
        int age;

        System.out.print("Enter your first name: ");
        name = keybd.nextLine();

        System.out.print("Enter your age " + name + ": ");
        age = keybd.nextInt();
        keybd.nextLine(); //eats up the rest of the line
    } //main
} //class
```

...so the previous program is better written like this

```
import java.util.Scanner;

/**
 * This program asks the user to enter their name and age/
 *
 * @author James Bond
 * @version v101
 */
public class AskForAge
{ //class

    public static void main (String[] args)
    { //main
        Scanner keybd = new Scanner(System.in);

        String name, s_age;
        int age;

        System.out.print("Enter your first name: ");
        name = keybd.nextLine();

        System.out.print("Enter your age " + name + ": ");
        s_age = keybd.nextLine();
        age = Integer.parseInt(s_age);
    } //main
} //class
```

```

/**
 * This program asks the user to enter their name and age/
 *
 * @author James Bond
 * @version v101
 */
public class AskForAge
{ //class

    public static void main (String[] args)
    { //main

        Scanner keybd = new Scanner(System.in);

        String name, s_age;
        int age;

        System.out.print("Enter your first name: ");
        name = keybd.nextLine();

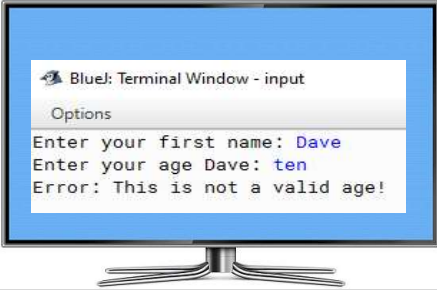
        System.out.print("Enter your age " + name + ": ");
        s_age = keybd.nextLine();

        try
        {
            age = Integer.parseInt(s_age);
            if(age > 0)
            {
                System.out.println("Your age number is " + age);
            }
        } catch (NumberFormatException ex)
        {
            System.out.println("Error: This is not a valid age!");
        }
    } //main
}

```

Data Validation

a try catch block can catch and deal with invalid user input



Primitive character literals are just a single character represented in single quotes.

```
char ch = 'c';
```

String literals are stored with double quotes. Strings are stored in memory as String objects and are written as a sequence of characters in double quotes.

```
String myName = new String("Bob");
```

or

```
String myName = "Bob";
```

There are a number of methods to compare String objects:

```

.equals ( )           s1.equals("hello")
                        returns true if contents of the strings are equal

.equalsIgnoreCase ( )   s1.equalsIgnoreCase("hello")
                        ignores uppercase/lowercase differences

.compareTo ( )         s1.compareTo("hello")
                        returns 0 if strings are equal,
                        negative number if s1 is less than "hello" alphabetically
                        positive number if s1 is greater than "hello"

```

The **.substring()** method creates a new String object by copying part of an existing String.

With one argument, copies from a starting index:

```

//                               20
String letters = "abcdefghijklmnopqrstuvwxyz";
String newString = letters.substring(20);           // "vwxyz"

```

With two arguments, copies from index up to next index

```

//           3456
String letters = "abcdefghijklmnopqrstuvwxyz";
String newString2 = letters.substring(3, 6);       // "def"

```


Method *replace* returns a new `String` object in which every occurrence of the first argument is replaced with the second.

// Can take char or String arguments

```
String newString = oldString.replace('x', 'y');
```

Method *toUpperCase* generates a new `String` with uppercase letters.
Method *toLowerCase* returns a new `String` object with lowercase letters.

```
String newString = oldString.toUpperCase();
```

Method *trim* generates a new `String` object that removes all whitespace characters that appear at the beginning or end.
Method *toArray* creates a new character array, a copy of the characters in the `String`.

```
char[] charArray = oldString.toCharArray();
```