

*** CS 106A MIDTERM SYNTAX REFERENCE ***

This document lists some of the common methods and syntax that you will use on the exam. For more, consult your textbook. (v1.3.2)

Karel the Robot (Karel reader Ch. 1-6)

```
public class Name extends SuperKarel { ... }
```

turnLeft(); turnRight(); turnAround();	rotates Karel 90° counter-clockwise, clockwise, or 180°
move();	moves Karel forward in current direction by one square
pickBeeper();	picks up a beeper if present on Karel's corner; else error
putBeeper();	places a beeper, if present in beeper bag; else error
frontIsClear(), frontIsBlocked()	Is there a wall in front of Karel?
leftIsClear(), leftIsBlocked()	Is there a wall to Karel's left (counter-clockwise)?
rightIsClear(), rightIsBlocked()	Is there a wall to Karel's right (clockwise)?
beepersPresent(), noBeepersPresent()	Are there any beepers on Karel's current corner?
beepersInBag(), noBeepersInBag()	Are there any beepers in Karel's beeper bag?
facingNorth(), notFacingNorth(), facingEast(), notFacingEast(), facingSouth(), notFacingSouth(), facingWest(), notFacingWest()	Is Karel facing north, south, east, or west?

Math (A&S 5.1)

```
double d = Math.pow(2, 5); // 32.0
```

```
Math.abs(n), Math.ceil(n), Math.floor(n), Math.log(n), Math.log10(n),  
Math.max(a, b), Math.min(a, b), Math.pow(b, e), Math.round(n), Math.sqrt(n),  
Math.sin(r), Math.cos(r), Math.tan(r), Math.toDegrees(r), Math.toRadians(d)
```

RandomGenerator (A&S 6.1)

```
RandomGenerator rg = RandomGenerator.getInstance();
```

<i>rg</i> .nextBoolean()	returns a random true/false result;
<i>rg</i> .nextBoolean(<i>probability</i>)	pass an optional probability from 0.0 - 1.0, or default to 0.5
<i>rg</i> .nextColor()	a randomly chosen Color object
<i>rg</i> .nextDouble(<i>min</i> , <i>max</i>)	returns a random real number between <i>min</i> and <i>max</i> , inclusive
<i>rg</i> .nextInt(<i>min</i> , <i>max</i>)	returns a random integer between <i>min</i> and <i>max</i> , inclusive

String (A&S Ch. 8)

```
String s = "hello";
```

s.charAt(<i>i</i>)	the character in this String at a given index
s.contains(<i>str</i>)	true if this String contains the other's characters inside it
s.endsWith(<i>str</i>)	true if this String ends with the other's characters
s.equals(<i>str</i>)	true if this String is the same as <i>str</i>
s.equalsIgnoreCase(<i>str</i>)	true if this String is the same as <i>str</i> , ignoring capitalization
s.indexOf(<i>str</i>)	first index in this String where given String begins (-1 if not found)
s.lastIndexOf(<i>str</i>)	last index in this String where given String begins (-1 if not found)
s.length()	number of characters in this String
s.replace(<i>s1</i> , <i>s2</i>)	a new string with all occurrences of <i>s1</i> changed to <i>s2</i>
s.startsWith(<i>str</i>)	true if this String begins with the other's characters
s.substring(<i>i</i> , <i>j</i>)	characters in this String from index <i>i</i> (inclusive) to <i>j</i> (exclusive)
s.toLowerCase()	a new String with all lowercase or uppercase letters
s.toUpperCase()	

Character/char (A&S Ch. 8)

```
char c = Character.toUpperCase(s.charAt(i));
```

Character.isDigit(<i>ch</i>), .isLetter(<i>ch</i>), .isLowerCase(<i>ch</i>), .isUpperCase(<i>ch</i>), .isWhitespace(<i>ch</i>)	methods that accept a char and return boolean values of true or false to indicate whether the character is of the given type
Character.toLowerCase(<i>ch</i>), .toUpperCase(<i>ch</i>)	accepts a character and returns lower/uppercase version of it

Scanner

```
Scanner input = new Scanner(new File("filename")); // scan an input file
Scanner tokens = new Scanner(string); // scan a string
```

<code>sc.next()</code> ,	<code>sc.nextLine()</code>	read/return the next token (word) or entire line of input as a string
<code>sc.nextInt()</code> ,	<code>sc.nextDouble()</code>	read/return the next token of input as an int or double
<code>sc.hasNext()</code> ,	<code>sc.hasNextLine()</code> ,	ask about whether a next token/line exists, or
<code>sc.hasNextInt()</code> ,	<code>sc.hasNextDouble()</code>	what type it is, without reading it
<code>sc.close()</code>		closes the scanner

ConsoleProgram

<code>public class Name extends ConsoleProgram { ... }</code>		
<code>readInt("prompt"),</code> <code>readDouble("prompt")</code>		Prompts/reprompts for a valid int or double, and returns it
<code>readLine("prompt");</code>		Prompts/reprompts for a valid String, and returns it
<code>readBoolean("prompt",</code> <code>"yesString", "noString");</code>		Prompts/reprompts for either yesString or noString (case-insensitive). Returns true if they enter yesString , false if they enter noString .
<code>promptUserForFile("prompt",</code> <code>"directory");</code>		Prompts for a filename, re-prompting until input is a file that exists in the given directory. Returns the full file path (" directory/filename ").
<code>println("text");</code>		Prints the given text to the console, followed by a newline ("n").
<code>print("text");</code>		Prints the given text to the console.

GraphicsProgram

<code>public class Name extends GraphicsProgram { ... }</code>		
<code>add(shape);</code>		displays the given graphical shape/object in the window
<code>add(shape, x, y);</code>		displays the given graphical shape/object in the window at x, y
<code>getElementAt(x, y)</code>		returns graphical object at the given x/y position, if any (else null)
<code>getHeight(), getWidth()</code>		the height and width of the graphical window, in pixels
<code>pause(ms);</code>		halts for the given # of milliseconds
<code>remove(shape);</code>		removes the graphical shape/object from window so it will not be seen
<code>setCanvasSize(w, h);</code>		sets canvas's onscreen size
<code>setBackground(color);</code>		sets canvas background color

Graphical Objects (A&S Ch. 9)

```
GRect rect = new GRect(10, 20, 50, 70);
```

<code>new GImage("filename", x, y)</code>	image from the given file, drawn at (x, y)
<code>new GLabel("text", x, y)</code>	text with bottom-left at (x, y)
<code>new GLine(x1, y1, x2, y2)</code>	line between points (x1, y1), (x2, y2)
<code>new GOval(x, y, w, h)</code>	largest oval that fits in a box of size w * h with top-left at (x, y)
<code>new GRect(x, y, w, h)</code>	rectangle of size w * h with top-left at (x, y)
<code>obj.getColor(), obj.getFillColor()</code>	returns the color used to color the shape outline or interior
<code>obj.getX(), obj.getY(),</code> <code>obj.getWidth(), obj.getHeight()</code>	returns the left x, top y coordinates, width, and height of the shape
<code>obj.move(dx, dy);</code>	adjusts location by the given amount
<code>obj.setBackground(IColor);</code>	sets overall window's background color
<code>obj.setFilled(booleAn);</code>	whether to fill the shape with color
<code>obj.setFillColor(IColor);</code>	what color to fill the shape with
<code>obj.setColor(IColor);</code>	what color to outline the shape with
<code>obj.setLocation(x, y);</code>	change the object's x/y position
<code>obj.setSize(w, h);</code>	change the objects width*height size

Colors

```
rect.setColor(Color.BLUE);
```

```
Color.BLACK, BLUE, CYAN, GRAY, GREEN, MAGENTA, ORANGE, PINK, RED, WHITE, YELLOW
Color name = new Color(r, g, b); // red, green, blue from 0-255
```

Mouse Events (A&S Ch. 10)

```
public void eventMethodName(MouseEvent event) { ...
events: mouseMoved, mouseDragged, mousePressed, mouseReleased, mouseClicked, mouseEntered, mouseExited
```

<code>e.getX(), e.getY()</code>	the x or y-coordinate of mouse cursor in the window
---------------------------------	---