*** CS 106A FINAL EXAM SYNTAX REFERENCE ***

Math (A&S 5.1)

`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)`

RandomGenerator (A&S 6.1)

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

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

User Input

`readInt(prompt)`	displays the given prompt, then reads and returns an integer value from the user
`readDouble(prompt)`	displays the given prompt, then reads and returns a double value from the user
`readLine(prompt)`	Displays the given prompt, then reads and returns a String from the user

String (A&S Ch. 8)

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

Character/char (A&S Ch. 8)

`Character.isDigit(ch), .isLetter(ch), .isLowerCase(ch), .isUpperCase(ch), .isWhitespace(ch)`	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(ch), .toUpperCase(ch)`	accepts a character and returns lower/uppercase version of it

BufferedReader (A&S 12.4)

```
BufferedReader br = new BufferedReader(new FileReader("filename"));
Exception handling: try/catch(IOException e)
```

`rd.readLine()`	return next line from file, or null if at the end
`rd.close()`	close file

Scanner

```
Scanner scan = new Scanner(new File("filename"));     OR     Scanner tokens = new Scanner(string);
```

`sc.next(), nextLine(), nextInt(), nextDouble()`	return next token/line of input
`sc.hasNext(), hasNextLine(), hasNextInt(), hasNextDouble()`	ask about next token/line without reading

Program, GraphicsProgram (*A&S* 2.6; Ch. 10)

`add(component/shape);` `add(component, region);`	displays the given component or graphical shape in the window; can optionally pass a region such as `SOUTH` or `EAST`
`addActionListeners();`	sets up your graphical program to hear action events on buttons
`addMouseListeners();`	sets up your graphical program to hear mouse events
`getHeight(), getWidth()`	the height or width of the graphical window, in pixels
`getElementAt(x, y)`	returns graphical object at the given x/y position, if any (else `null`)
`getElementAt(x1, y1, x2, y2, x3, y3, x4, y4)`	returns graphical object at any of the given x/y pairs, if any (else `null`)
`pause(ms);`	halts for the given # of milliseconds

`remove(component/shape);`	removes the component or graphical shape from the window
`setSize(w, h);`	sets the console window's onscreen size

Graphical Objects (A&S Ch. 9)

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

Colors

```
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`

`e.getButton()`	which mouse button was pressed, if any
`e.getX(), e.getY()`	the x or y-coordinate of mouse cursor in the window

Action Events (A&S Ch. 10)

```
public void actionPerformed(ActionEvent event) { ... }
```

`e.getActionCommand()`	a string representing the event that occurred
`e.getSource()`	the component that caused the event

Swing Graphical Components (*A&S* 10.6)

`JButton,JCheckBox,JColorChooser,JFileChooser,JLabel,JRadioButton,JSlider,JStringList,JTextArea,JTextField`

`c.addActionListener(listener)`	sets up component to notify listener when action events occur
`c.isEnabled(), c.setEnabled(boolean)`	get/set whether the component is able to be clicked on
`c.isSelected(), c.setSelected(boolean)`	get/set whether the component is currently checked (checkbox/radio)
`c.getText(), c.setText(String)`	get/set the text being displayed on the component

ArrayList (A&S 11.8)

```
ArrayList<Integer> list = new ArrayList<Integer>();
```

`L.add(value);` `L.add(index, val);`	append to end of list; or insert before index, shifting right
`L.clear();`	removes all elements of list
`L.contains(value)`	`true` if value is in the list
`L.equals(L2)`	`true` if same elements
`L.get(index)`	returns value at given index
`L.indexOf(value)` `L.lastIndexOf(val)`	first/last index where given value is found (or -1 if not found)
`L.isEmpty()`	`true` if the list has no elements
`L.remove(index);`	removes value at given index, shifting subsequent values left
`L.remove(val);`	removes first occurrence of value
`L.set(index, val);`	replaces value at given index
`L.size()`	number of elements in the list
`L.toString()`	string representation of list such as `"[10, -2, 43]"`

HashMap (A&S 13.2)

```
HashMap<String, Double> gpa = new HashMap<String, Double>();
```

`M.put(key, value);`	adds a pair between the given key and value, replacing old pair for that key
`M.clear();`	removes all elements of map
`M.containsKey(key)`	returns `true` if the given key is a key of a pair in this map
`M.equals(map2)`	`true` if same elements
`M.get(key)`	returns value paired with key, or `null`
`M.isEmpty()`	`true` if the map has no pairs
`M.remove(key);`	removes pair for the given key, if there is one; does nothing if not
`M.keySet()`	a collection of all keys in the map
`M.values()`	collection of all values in map
`M.size()`	returns number of pairs in map
`M.toString()`	returns a string representation such as `"{a=b, c=d, e=f}"`