

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

This document lists some of the common methods and syntax that you will use on the exam.

Math (A&S 5.1)

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

| |
|---|
| Math.abs(<i>n</i>), Math.ceil(<i>n</i>), Math.floor(<i>n</i>), Math.log(<i>n</i>), Math.log10(<i>n</i>), Math.max(<i>a</i> , <i>b</i>), Math.min(<i>a</i> , <i>b</i>), Math.pow(<i>b</i> , <i>e</i>), Math.round(<i>n</i>), Math.sqrt(<i>n</i>), Math.sin(<i>r</i>), Math.cos(<i>r</i>), Math.tan(<i>r</i>), Math.toDegrees(<i>r</i>), Math.toRadians(<i>d</i>) |
|---|

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";
```

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

Integer/int (A&S Ch. 8)

```
Int num = Integer.parseInt("106");
```

| | |
|--------------------------|--|
| Integer.parseInt(String) | accepts a numerical String and returns the value as an int |
|--------------------------|--|

Scanner

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

| | |
|--|--|
| <i>sc</i> .next(), <i>sc</i> .nextLine() | read/return the next token (word) or entire line of input as a string |
| <i>sc</i> .nextInt(), <i>sc</i> .nextDouble() | read/return the next token of input as an int or double |
| <i>sc</i> .hasNext(), <i>sc</i> .hasNextLine(), <i>sc</i> .hasNextInt(), <i>sc</i> .hasNextDouble() | ask about whether a next token/line exists, or what type it is, without reading it |
| <i>sc</i> .useDelimiter(String) | set the character(s) on which the scanner breaks input into tokens |
| <i>sc</i> .close() | closes the scanner |

Program, ConsoleProgram, GraphicsProgram

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

| | |
|---------------------|--|
| <code>init()</code> | executes before window appears; use to set up graphical components |
| <code>run()</code> | executes after window appears; use for animation loops, file loading, etc. |

ConsoleProgram

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

| | |
|--|--|
| <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 <i>yesString</i> or <i>noString</i> (case-insensitive). Returns true if they enter <i>yesString</i> , false if they enter <i>noString</i> . |
| <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 (" <i>directory/filename</i> "). |
| <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

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

| | |
|--|--|
| <code>add(shape), 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 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.setFilled(boolea);</code> | whether to fill the shape with color |
| <code>obj.setFillColor(Color);</code> | what color to fill the shape with |
| <code>obj.setColor(Color);</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 object's width and height |
| <code>Label.setLabel(String);</code> | changes the text that a GLabel displays |
| <code>Label.getAscent(), Label.getDescent()</code> | returns a GLabel's ascent or descent from the baseline |
| <code>new GImage("filename", x, y)</code> | image from the given file, drawn at (x, y) |
| <code>new GImage(pixelArray)</code> | image from the given 2D array of int pixels |
| <code>image.getPixelArray(),</code> <code>setPixelArray(a)</code> | return/set 2D array of ints representing pixels of the image |
| <code>GImage.getRed(px), getGreen(px),</code> <code>getBlue(px)</code> | returns the individual red/green/blue components of a given int pixel |
| <code>GImage.createRGBPixel(r, g, b)</code> | creates and returns an int pixel with the given r/g/b values |
| <code>GImage.createRGBPixel(r, g, b, a)</code> | creates and returns an int pixel with the given r/g/b/alpha values |

Colors

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

| |
|---|
| Color.BLACK, BLUE, CYAN, GRAY, GREEN, MAGENTA, ORANGE, PINK, RED, WHITE, YELLOW Color <i>name</i> = new Color(<i>r</i> , <i>g</i> , <i>b</i>); // 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.getX(),    e.getY()                      the x or y-coordinate of mouse cursor in the window
```

Array (A&S Ch. 11)

```
int[] arr = new int[5];                      int[][] pixels = new int[5][2];
```

| | |
|---|---|
| new <i>type</i> [<i>length</i>] | creates a new 1D array of the given type and length |
| new <i>type</i> [<i>rows</i>][<i>cols</i>] | creates a new 2D array of the given type and number of rows and cols |
| <i>arr</i> [<i>i</i>], <i>arr</i> [<i>i</i>][<i>j</i>], ... | returns the element at index <i>i</i> , index (<i>i</i> , <i>j</i>), etc. |
| <i>arr.length</i> | returns the length of the array |
| Arrays.toString(<i>arr</i>) | returns a string representing the array, such as “[10, 30, -25, 17]” |
| Arrays.sort(<i>arr</i>) | sorts the elements in place (no return value) |
| Arrays.equals(<i>arr1</i> , <i>arr2</i>) | returns true if the arrays contain the same elements in the same order |
| Arrays.fill(<i>arr</i> , <i>value</i>) | sets every element to the given value |
| Arrays.deepToString(<i>arr</i>) | returns a string representing the multidimensional array, such as “[[0, 1, 2], [1, 2, 3], [2, 3, 4]]” |
| Arrays.deepEquals(<i>arr1</i> , <i>arr2</i>) | returns true if the multidimensional arrays contain the same elements in the same order. |

ArrayList (11.8)

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

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

HashMap (13.2)

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

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

// *collection* is a HashMap key/value set, array, or ArrayList
for (*type name* : *collection*) { ...

Interactors (A&S 10.5-10.6)

JButton button = new JButton("Click me!");

| | |
|--|---|
| new JButton("text") | button displaying the given text |
| addActionListeners() | sets up program to hear action events on all added buttons |
| new JLabel("text") | label displaying the given text |
| new JTextField(width) | text field with the given width (in characters) |
| textField .addActionListener(this) | sets up program to hear an action event when ENTER key typed |
| .getText(), .setText(text) | get/set the text being displayed in the button/label/text field |
| add(component , region) | adds the given interactor in the given window region (e.g. SOUTH) |
| .setActionCommand("text"), .getActionCommand() | gets/sets the action command associated with an interactor. |

public void **actionPerformed**(ActionEvent event) { ...

| | |
|------------------------------|--|
| e .getActionCommand() | action command of the triggered interactor (e.g. text of clicked button) |
| e .getSource() | the triggered component/interactor itself |