# Preparing for AP Computer Science A Exam

**FORMAT OF THE EXAM:**
Section I (50%) - Multiple Choice - 40 questions - 75 minutes (1.9 min. per question)

Section II (50%) - Free Response - 4 questions - 105 min. (26 min. per question)

YOU ARE PREPARED, but remember the **TIPS** we talked about:

**MULTIPLE CHOICE QUESTIONS**:
Make sure you write down and trace code!

- Your Reference Guide could also be helpful with this section, especially Appendix A which has informat important Java methods.
- Save long, wordy questions for the end and come back to them later.
- Answer ALL questions. You will not be penalized for wrong answers this year.
- Read carefully, but quickly.
- Read the question before taking time to look at the code

**FREE RESPONSE**:
- Use your Reference Guide
- If you feel most comfortable with GridWorld, try that question first.
- ALWAYS make sure that you return the correct data type if the method is not void. If you don't know hc problem, at least create a variable of that data type and return it at the bottom of the method.
- If the method has a parameter that is an array or an array list, you will probably need to loop through i If you don't know the solution then at least do that.
- Always use previously defined methods (even if you know you haven't coded it properly) instead of tryir the functionality in another method.
- Do all that you can. You receive points for each successful part that they are grading.

**Click here for other useful tips**

Be prepared for the **CONTENT** on the exam:
- Review the **Topic Outline** (page 8 of Course Description)
- Review the **Java Subset** (Appendix A)

**PRACTICE** by using these websites:
- Prior 2010, 2011, 2012, 2013, AP Bowl GT MC questions.
- **2012 Free Response Questions** and possible **Solutions**
- **2011 Free Response Questions** and possible **Solutions**
- **College Board free response questions**
- http://pages.eimacs.com/eimacs/signin IMACS (multiple choice questions)
- **CodingBat (free response practice)**

## Standard Java Library Methods Required for AP CS A

**Accessible Methods from the Java Library That May Be Included on the Exam**

**class java.lang.Object**

- boolean equals(Object other)
- String toString()

**class java.lang.Integer**

- Integer(int value)
- int intValue()
- Integer.MIN_VALUE // minimum value represented by an int
- Integer.MAX_VALUE // maximum value represented by an int

**class java.lang.Double**

- Double(double value)
- double doubleValue()
-

**class java.lang.String**

- int length()
- String substring(int from, int to)

  // returns the substring beginning at from

  // and ending at to-1
- String substring(int from)

  // returns substring(from, length())
- int indexOf(String str)

  // returns the index of the first occurrence of str;

  // returns -1 if not found
- int compareTo(String other)

  // returns a value < 0 if this is less than other

  // return a value = 0 if this is equal to other

  // return a value > 0 if this is greater than other

**class java.lang.Math**

- static int abs(int x)
- static double abs(double x)
- static double pow(double base, double exponent)
- static double sqrt(double x)
- static double random()

  // returns a double in the range [0.0, 1.0)

**class java.util.List<E>**

- int size()
- boolean add(E obj)

  // appends obj to the end of list; returns true
- void add(int index, E obj)

  // inserts obj at position index (0<= index <= size),

  // moving elements at position index and higher

  // to the right (adds 1 to their indices) and adjusts size
- E get(int index)
- E set(int index, E obj)

  // replaces the element at position index, with obj

  //returns the element formerly at the specified position
- E remove(int index)

  // removes element from position index, moving elements

  // at position index + 1 and higher to the left

  // (subtracts 1 from their indices) and adjusts size

  // returns the element formerly at the specified position

class java.util.ArrayList implements java.util.List

## Summary

| Tested in A exam | |
|---|---|
| `int, double, boolean Integer.MAX_VALUE, Integer.MIN_VALUE` | |
| `+ , -, *, /, %, ++, --` | |
| `=, +=, -=, *=, /=, %=` | |
| `==, !=, <, <=, >, >=` | |
| `&&, \|\|, !` and short-circuit evaluation | |
| (`int`), (`double`) | |
| String concatenation | |
| Escape sequences `\", \\, \n` inside strings | |
| `System.out.print, System.out.println` | |
| 1-dimensional arrays<br>2-dimensional rectangular arrays; | |
| `if, if/else, while, for, enhanced for, return` | |
| Modify existing classes, design classes | |
| `public` classes, `private` instance variables, `public` or `private` methods or constants | |
| @param, @return | |
| `static` class variables | |
| `static` methods | |
| `null, this, super, super.method(args)` | |
| Constructors and initialization of `static` variables | |
| Understand inheritance hierarchies. Design and implement subclasses. Modify subclass implementations and implementations of interfaces. | |
| Understand the concepts of abstract classes and interfaces. | |
| Understand `equals, ==,` and `!=` comparison of objects `String.compareTo` | |
| Conversion to supertypes and (`Subtype`) casts | |

| | |
|---|---|
| Package concept, `import packageName.ClassName;` | |
| Exception concept, common exceptions | |
| `String, Math, Object, List, ArrayList` | |
| Wrapper classes (`Integer, Double`) | |