

APCSA Exam

Please read this document to review what you should know for the exam:

<https://secure-media.collegeboard.org/digitalServices/pdf/ap/ap-computer-science-a-java-subset.pdf>

You will have access to a reference sheet during the exam:

https://apstudents.collegeboard.org/ap/pdf/ap-computer-science-a-java-quick-reference_0.pdf

I assigned multiple practice questions in the College Board Classroom. Please check the assignments and quizzes section.

CSAwesome has all the AP material including exercises. If you would like to review:

<https://runestone.academy/ns/books/published/csawesome/index.html>

Common mistakes

Here is a list of the most common mistakes students have had in exams:

Expressions

Remember the order of operations. Be careful with this types of expressions (integers and strings):

`2 + 4 + "5" + 6 => "656"`

`5%2 => remainder`

Random numbers

Please make sure you study how to generate random numbers using `Math.random()` and the `Random` class.

```
// Generates a number between 10(inclusive) and 20(exclusive)
int min = 10;
int max = 20;
int range = max - min + 1;
// with Random class
Random rand = new Random();
System.out.println(rand.nextInt(range) + 10)
// with Math.random
System.out.println((int) (Math.random() * range) + min);
```

Strings

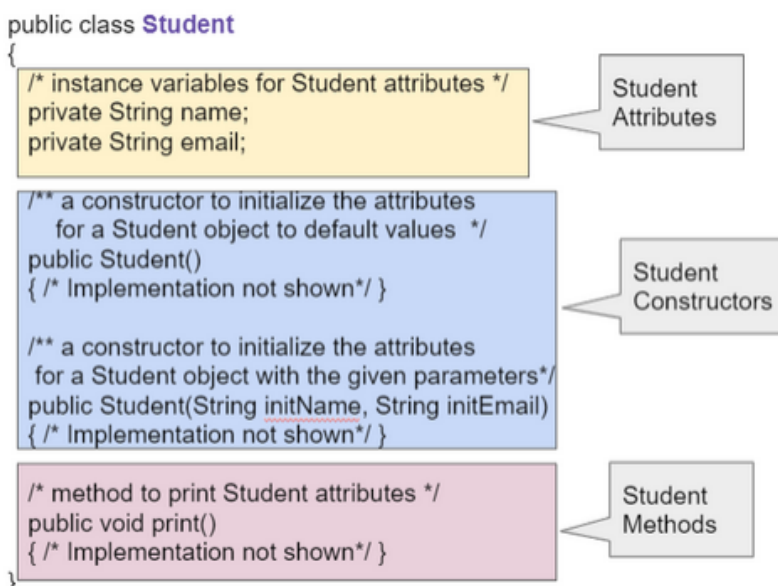
Review the most common string methods: length, substring, indexOf, equals, compareTo.

When you compare strings, you must use equals. DO NOT use ==.

Objects Classes

Constructors: Initialize attributes in newly created objects. They have the same name as the class. You may have more than one constructor (overloading constructors).

Instance variables: The instance variables of a class should be private.



ArrayList

Review all methods.

`remove(index)`: It removes and returns the element at the given index.

`set(index, element)`: Returns the element previously at the specified position.

Be careful when you are traversing an ArrayList and delete elements at the same time, you will skip some values. Example:

Assume that `nums` is an ArrayList with these values `[0, 0, 4, 2, 5, 0, 3, 0]`. What will `nums` contain after executing the following code:

```

public static void numQuest(ArrayList<Integer> nums)
{
    int k = 0;
    int zero = 0;
    while (k < nums.size())
    {
        if (nums.get(k).equals(zero))
            nums.remove(k);
        k++;
    }
    System.out.println(nums);
}

```

nums will contain [0, 4, 2, 5, 3], as you can see not all zeros were removed.

2D Arrays

`int[][] myArray = new int [2][3];` => First number is for rows, second one is for columns.

Inheritance

The keyword **extends** is used to establish an **inheritance** relationship between a **subclass** and a **superclass**. A class can extend only one superclass.

Subclasses do not have access to the private instance variables in a superclass that they extend.

A superclass constructor must be called from the first line of a subclass constructor by using the keyword `super` and passing appropriate parameters. If there is no explicit call to `super` an implicit call to `super()` will be added by the Java compiler.

The `toString` method is commonly overridden. A subclass can override `toString` but in its new `toString` method, it can call `super.toString()` to get a string to which it can add its own instance variables.

Polymorphism: At compile time, java compiler makes sure all methods are implemented in the declared type class of an object, While at running time the method executed will be the one in the class used to instantiate the object, if it exists there. Otherwise the methods executed will be the one in the declared type class.



Static Methods

- It belongs to the class rather than the object of a class.
- It can be invoked without the need for creating an instance of a class.
- It can access static data members and can change the value of it.

Non-Static methods

- It does not have the keyword static before the name of the method.
- It belongs to an object of the class and you have to create an instance of the class to access it.
- It can access any static method and any static variable without creating an instance of the class.

Static vs. Non-Static Variables

STATIC VARIABLE	NON STATIC VARIABLE
Static variables can be accessed using class name	Non static variables can be accessed using instance of a class
Static variables can be accessed by static and non static methods	Non static variables cannot be accessed inside a static method.
Static variables reduce the amount of memory used by a program.	Non static variables do not reduce the amount of memory used by a program
Static variables are shared among all instances of a class.	Non static variables are specific to that instance of a class.
Static variable is like a global variable and is available to all methods.	Non static variable is like a local variable and they can be accessed through only instance of a class.

```
public class Client{
    private static int points = 0;
    private int cashBack = 0;
    public Client(){
        points += 100;
        cashBack += 10;
    }
    public int getPoints(){
        return points;
    }
    public int getCashBack(){
        return cashBack;
    }
}
```

```
Client c1 = new Client();
Client c2 = new Client();
Client c3 = new Client();
c1.getPoints(); => 300
c2.getPoints(); => 300
c3.getPoints(); => 300
c1.getCashBack(); => 10
c2.getCashBack(); => 10
c3.getCashBack(); => 10
```