

## CSCI 161 Final Exam Online Problem

This portion of the exam is open book and notes (your own only, of course). You may not consult anyone other than the instructor during this portion of the exam. You MAY use any code that you have previously developed in order to complete this programming problem. Please read the problem in its entirety before beginning.

**Description:** Develop a Java program named *FinalExam.java* that:

- a. Reads top grossing musical artist sales figures from a file with each line of the file containing year, gross sales and the top grossing artist name; storing the data in the arrays named *years*, *sales*, and *artists*, respectively.
- b. Prints out all data (year, gross sales, artist) in a formatted fashion as well as outputting of the top-grossing artist.

**Input Specification:** Your program will need to read the following file from the project directory (current working directory), with the file described as follows:

**artistsales.txt** – This file contains year, gross sales and top artist for said year in the following format:

```
1992 4700000 Billy Ray Cyrus
1993 5460000 Whitney Houston
1994 4934000 Elton John
1995 7020000 Hootie and the Blowfish
1996 7380000 Alanis Morissette
1997 5302000 Spice Girls
1998 9338000 James Horner
1999 9446000 Backstreet Boys
2000 9936000 NSYNC
2001 4813000 Linkin Park
2002 7608000 Eminem
2003 6536000 50 Cent
2004 7979000 Usher
2005 4968606 Mariah Carey
2006 3719000 High School Musical (soundtrack)
2007 3699000 Josh Groban
2008 2880000 Lil Wayne
2009 3217000 Taylor Swift
2010 3415000 Eminem
2011 5824000 Adele
2012 4410000 Adele
2013 2430000 Justin Timberlake
2014 3661000 Taylor Swift
2015 8008000 Adele
2016 4140000 Drake
2017 2764000 Ed Sheeran
2018 1491000 Various Artists
2019 1085000 Taylor Swift
2020 1276000 Taylor Swift
2021 1464000 Adele
```

You should cut and paste the contents above into your own file and make sure the file is in the current working directory of your program and is named **artistsales.txt** (exactly, all lower case).

**Output Specification:** Your program should output two sections of information. The *first* section includes the formatted list for all the top grossing artists for each year in the format below. Please note the special formatting of the sales as those figures will always be in millions of dollars for the sample data provided. Figures should be formatted with one decimal place, e.g. \$4.7M).

Then, the *second* section lists the highest grossing artist from the list.

For the format of both sections, see the following output example:

YEAR	GROSS	ARTIST
1992	\$4.7M	Billy Ray Cyrus
1993	\$5.5M	Whitney Houston
1994	\$4.9M	Elton John
1995	\$7.0M	Hootie and the Blowfish
1996	\$7.4M	Alanis Morissette
1997	\$5.3M	Spice Girls
1998	\$9.3M	James Horner
1999	\$9.4M	Backstreet Boys
2000	\$9.9M	NSYNC
2001	\$4.8M	Linkin Park
2002	\$7.6M	Eminem
2003	\$6.5M	50 Cent
2004	\$8.0M	Usher
2005	\$5.0M	Mariah Carey
2006	\$3.7M	High School Musical (soundtrack)
2007	\$3.7M	Josh Groban
2008	\$2.9M	Lil Wayne
2009	\$3.2M	Taylor Swift
2010	\$3.4M	Eminem
2011	\$5.8M	Adele
2012	\$4.4M	Adele
2013	\$2.4M	Justin Timberlake
2014	\$3.7M	Taylor Swift
2015	\$8.0M	Adele
2016	\$4.1M	Drake
2017	\$2.8M	Ed Sheeran
2018	\$1.5M	Various Artists
2019	\$1.1M	Taylor Swift
2020	\$1.3M	Taylor Swift
2021	\$1.5M	Adele

TOP GROSSING ARTIST

-----  
2000        \$9.9M    NSYNC

NOTE: Your output, including all headings, hyphens/dashes, spaces between columns, etc. should match identically.

**Hints and Suggestions:** Following are several hints and suggestions to aid you the development of your program:

1. **Use global array variables:** Make your life easier and start with the arrays that you need as global class variables declared at the top of your program instead of in the main method (thus avoiding the need to pass them around.) To review, the arrays that you need are:

```
/** Class Variables */
public static int[] years;           // read from file
public static int[] sales;          // read from file
public static String[] artists;     // read from file
```

2. **Develop methods wisely** – If all your arrays are in the global space there is no need to pass them to any of your methods. Other than that, helpful methods *might* include (copy and paste if you wish):

```
// Returns number of lines in a supplied file
public static int numLines(String fname)

// Figures out size of file to read, sizes arrays to be equal
// to number of lines in file, reads file and fills years,
// sales and artists arrays
public static void readArtistSalesFile(String fname)

// Given artist gross sales figure returns a string
// in the format:  $%.1fM
// for millions of dollars.
public static String formatSales(int grossSales)

// Prints information for a single artist sales given its index
public static void printArtist(int idx)

// Prints header and then all artist sales
public static void printAllArtists()

// Print header then find and print info for highest grossing artist
public static void printTopArtist()
```

Note: If your program utilizes methods like those above calling the proper methods, in order, should be straightforward and your main method rather simple with few statements (three to be exact).

3. **Iteratively Develop your Program:** Build your program in small increments, testing as you go, so that you always have something that compiles. A program that is nearly complete and cleanly compiles will receive more points than a complete program that does not compile! Remember that you may submit early versions of the program as many times as you like!! Submit your *FinalExam.java* program via AutoLab as the *FinalExam* exam.

**Grading:** This portion of your exam is worth 25 points with the following grading rubric used. Your program earns percentages towards that total 25 points as follows:

- +80% - Compiles, runs, and **artistsales.txt** file properly read into *years*, *sales* and *artists* arrays.
- +10% - Determines and prints highest grossing artist.
- +5% - Prints artist sales stats in properly formatted columnar data.
- +5% - Prints sales figures in *friendly* millions format.