



# JAVA PROGRAMMING

## (340)

### REGIONAL – 2017

**Production Portion:**

Program 1: School Vending Machine \_\_\_\_\_ (400 points)

***TOTAL POINTS*** \_\_\_\_\_ (***400 points***)

**Judge/Graders: Please double check and verify all scores and answer keys!**

Property of Business Professionals of America.  
May be reproduced only for use in the Business Professionals of America  
*Workplace Skills Assessment Program* competition.



You will have 90 minutes to complete your work.

Your name and/or school name should NOT appear on any work you submit for grading.

Create a folder on the flash drive provided using your contestant number as the name of the folder. Copy your entire solution/project into this folder. You must submit your entire solution/project so that the graders may open your project to review the source code. You must ensure that the files required to run your program are present and will execute on the flash drive provided. Note that the flash drive letter may not be the same when the program is graded as it was when you created the program. It is recommended that you use relative paths rather than absolute paths to ensure that the program will run regardless of the flash drive letter. The graders will *not* compile or alter your source code to correct for this. Submissions that do *not* contain source code will *not* be graded.

Assumptions to make when taking this assessment:

- No external files are required for this project.

Development Standards:

- Your Code must use a consistent variable naming convention.
- All subroutines, functions, and methods must be documented with comments explaining the purpose of the method, the input parameters (if any), and the output (if any). If you create a class, then you must use Javadoc comments.



Your application will be graded on the following criteria:

### **Solution and Project**

- The project is present on the flash drive \_\_\_\_\_ 10 points
- The projects main class is named SchoolVending \_\_\_\_\_ 10 points

### **Program Execution**

- The program runs from the USB flash drive \_\_\_\_\_ 15 points

If the program does *not* execute, then the remaining items in this section receive a score of zero.

- The program runs and produces the initial menu screen \_\_\_\_\_ 20 points
- The program adds the appropriate item cost total sale value based on the input option \_\_\_\_\_ 20 points
- The program keeps track of the number items selected \_\_\_\_\_ 20 points
- The program resets the transaction to all zeros when the “Clear Transaction” option is selected \_\_\_\_\_ 20 points
- The program clears the screen every time the menu is displayed \_\_\_\_\_ 20 points
- The program displays an error message and pauses for approximately two seconds if an illegal entry is made. \_\_\_\_\_ 40 points
- The “Current Sales Total” is formatted to have a dollar sign and shows only two decimal places. \_\_\_\_\_ 30 points
- The program displays the final output screen when appropriate \_\_\_\_\_ 30 points
- Output looks like the examples given \_\_\_\_\_ 35 points

### **Source Code Review**

- The source code is properly commented
  - A comment containing the contestant number is present \_\_\_\_\_ 10 points
  - Methods and code sections are commented \_\_\_\_\_ 20 points
- A method exists to display the menu and validate entry \_\_\_\_\_ 40 points
- A method exists to perform the sleep/pause function \_\_\_\_\_ 20 points
- A method exists to perform the clear screen function \_\_\_\_\_ 20 points
- Code uses a consistent variable naming convention \_\_\_\_\_ 10 points
- Methods are used appropriately \_\_\_\_\_ 10 points

**Total Points: 400 points**



## School Vending Prototype

Professional Business Associates has been asked to develop the prototype software to be used in a school vending machine. These vending machines will be used to sell school supplies. The concept of the vending machine is that it will be able to sell multiple items per transaction. The user will make choices via a menu system and the machine will tell the user how much the entire transaction will cost.

### Menu:

The menu will display the input option, item description, and item price. See the example below for the list of items to be offered and their cost. Due to the hardware design of the display screen on the vending machine, the menu must be must look identical to the example on the next page.

### Input:

Since this is a prototype design, the program's input will be received from the keyboard. All input should be validated for validity. The program should be able to handle input of both upper and lower case. For example the input of "a" and the input of "A" should be considered the same. The program should continue to take input from the user until the user enters either an upper or lower case "H" or an upper or lower case "X".

### Output:

The output be as close to the examples below as possible.

### Requirements:

1. You must create an application with the main class named SchoolVending.
2. Your contestant number must appear as a comment at the top of the main source code file.
3. The program should use **at least** three methods besides the main. The first method is the menu. It should clear the current screen and it should print out the current number of items selected and the cost of the transaction (see Menu Screen figure on the next page). The menu should validate that only the appropriate options have been entered. If an invalid entry is entered, the program should print a message to the user and pause for approximately 2 seconds before redisplaying the menu (see Illegal Entry figure on page 6).
4. The second method should handle the "pause/sleep" for the invalid entry.
5. The third method should handle the "clear screen" operation.
6. If you order all six items on the menu and select the "Exit and Pay" option, your program should print the final output screen (see Final Output figure on page 6).



```
Blue: Terminal Window - Regional_17
Options
Current Sales
Items Selected: 0
Current Sales Total: $0.00

School Vending Menu
A: Notebook Paper - $2.00   B: Mechanical Pencils (3 Pack) - $1.00
C: 3 Ring Binder - $3.00   D: Pens (3 Pack) (Black,Red,Blue - $1.00
E: Folder - $0.75         F: Highlighter - $0.50
G: Clear Transaction      H: Exit and pay
X: Cancel and Exit
Please Choose from the menu printed above:
```

Menu Screen



```
BlueJ: Terminal Window - Regional_17
Options
Current Sales
Items Selected: 0
Current Sales Total: $0.00

School Vending Menu
A: Notebook Paper - $2.00   B: Mechanical Pencils (3 Pack) - $1.00
C: 3 Ring Binder - $3.00   D: Pens (3 Pack) (Black,Red,Blue - $1.00
E: Folder - $0.75         F: Highlighter - $0.50
G: Clear Transaction      H: Exit and pay
X: Cancel and Exit
Please Choose from the menu printed above: z
Illegal Entry!
```

**Illegal Entry**

```
BlueJ: Terminal Window - Regional_17
Options
Final Sales
Items Selected: 6
Current Sales Total: $8.25
```

**Final Output**



Code For Key:

```
import java.util.Scanner;
import java.text.NumberFormat;
/**
 * Write a description of class SchoolVending here.
 *
 * @author (your contestant number)
 * @version (a version number or a date)
 */
public class SchoolVending
{

    public static void main(String[] args)
    {
        //Variable Declaration
        String Choice;
        int items = 0;
        double subTotal = 0;
        double grandTotal = 0;
        double tax = 0;

        //Continue to call the menu until exit option
        do
        {
            Choice = Menu(items,subTotal); // Get the value of choice by calling menu
            // if statments to handle the user's input.
            if(Choice.equalsIgnoreCase("A"))
            {
                items ++;
                subTotal += 2;
            }
            else if(Choice.equalsIgnoreCase("B"))
            {
                items ++;
                subTotal += 1;
            }
            else if(Choice.equalsIgnoreCase("C"))
            {
                items ++;
                subTotal += 3;
            }
            else if(Choice.equalsIgnoreCase("D"))
            {
                items ++;
                subTotal += 1;
            }
        }
    }
}
```



```
        else if(Choice.equalsIgnoreCase("E"))
        {
            items ++;
            subTotal += .75;
        }
        else if(Choice.equalsIgnoreCase("F"))
        {
            items ++;
            subTotal += .5;
        }
        else if(Choice.equalsIgnoreCase("G"))
        {
            items =0;
            subTotal =0;
        }
        else if(Choice.equalsIgnoreCase("H"))
        {
            FinalOutput(items,subTotal); //calls FinalOutput with current sales
        }
        else
        {
            items =0;
            subTotal =0;
            FinalOutput(items,subTotal); //calls FinalOutput with zeros
        }

    }
    while(!Choice.equalsIgnoreCase("H")&&!Choice.equalsIgnoreCase("X"));

}

/**
 * Displays the current sales total and menu. Validates all input while ignoring case
 * pre: Items and Sub
 * post: returns the user's validated option
 */
public static String Menu(int items, double Sub)
{
    String userInput;
    NumberFormat formatter = NumberFormat.getCurrencyInstance(); //used to format current
sales total
    Scanner input = new Scanner(System.in); //takes in user input
    do
    {
        CLS(); //calls the clear screen method
        //Current Sale
        System.out.println("                Current Sales");
```





```
System.out.println("Items Selected: " + items);
System.out.print("Current Sales Total: ");
System.out.println(formatter.format(Sub));
System.out.println();
System.out.println();
// Menu Screen
System.out.println("          School Vending Menu" );
System.out.println("A: Notebook Paper - $2.00  B: Mechanical Pencils (3 Pack) - $1.00");
System.out.println("C: 3 Ring Binder - $3.00  D: Pens (3 Pack)(Black,Red,Blue - $1.00");
System.out.println("E: Folder      - $0.75  F: Highlighter          - $0.50");
System.out.println("G: Clear Transaction      H: Exit and pay");
System.out.println("X: Cancel and Exit");
System.out.print("Please Choose from the menu printed above: ");
userInput = input.nextLine();
//validates user input
if(!userInput.equalsIgnoreCase("A")) &&(!userInput.equalsIgnoreCase("B")) &&
(!userInput.equalsIgnoreCase("C")) && (!userInput.equalsIgnoreCase("D")) &&
(!userInput.equalsIgnoreCase("E")) && (!userInput.equalsIgnoreCase("F")) &&
(!userInput.equalsIgnoreCase("G")) && (!userInput.equalsIgnoreCase("H")) &&
(!userInput.equalsIgnoreCase("X")))
{
    System.out.println("Illegal Entry!");
    Sleep(2);//calls the sleep method (2 seconds)
    CLS();//calls the clear screen method
}

}

while(!userInput.equalsIgnoreCase("A")) &&(!userInput.equalsIgnoreCase("B")) &&
(!userInput.equalsIgnoreCase("C")) && (!userInput.equalsIgnoreCase("D")) &&
(!userInput.equalsIgnoreCase("E")) && (!userInput.equalsIgnoreCase("F")) &&
(!userInput.equalsIgnoreCase("G")) && (!userInput.equalsIgnoreCase("H"))&&
(!userInput.equalsIgnoreCase("X")));

return (userInput.toUpperCase());
}

/**
 * Clear Screen Method
 * pre: none
 * post: clears the screen
 */
public static void CLS()
{
    System.out.print("\u000C");
}

/**
```



```
* pauses the program
* pre: number of seconds to pause
* post: pauses the program
*/
public static void Sleep(int seconds)
{
    try {
        Thread.sleep(1000*seconds);           //1000 milliseconds is one second.
    } catch(InterruptedException ex) {
        Thread.currentThread().interrupt();
    }
}

/**
 * Displays the Final Output screen
 * pre: number of items, and subtotal of sales
 * post: none
 */
public static void FinalOutput(int items, double subTotal)
{
    NumberFormat formatter = NumberFormat.getCurrencyInstance();

    CLS();
    System.out.println("          Final Sales");
    System.out.println("Items Selected: " + items);
    System.out.print("Current Sales Total: ");
    System.out.println(formatter.format(subTotal));

}
}
```