St. Edmund Preparatory High School
Brooklyn, NY

AP Computer Science
Mr. A. Pinnavaia
Summer Assignment

Name: _____

I know it has been about 7 months since you last thought about programming. It's ok. I wouldn't want to think about code at the beach either. Since it has been a little while, let's see what you remember from Advanced Programming Principles. To the best of your ability, describe the concepts listed below.

1. Algorithm

2. Low Level Languages vs. High Level Languages

3. Complier

4. Variables

5. Variable Type

6. Variable Declarations

7. Output

8. Arithmetic Operators

9. Relational Operators

10. Conditional Statements (*C++: IF and IF…Else*)

11. AND statements

12. OR statements

13. Loop

14. Functions

15. Classes

16. Objects

St. Edmund Preparatory High School
Brooklyn, NY

Many of the basic programming concepts exist in every programming language. There are many things you already know what they do and/or how they work. This year you are going to learn some different ways to write those basic programming concepts.

Below is a sample Java program from your textbook (pg. 15). While you may not understand all of the code in this program and what the specific instructions are, it is very likely you can tell me what the parts of this program are doing. Label the parts of this program and, based on your knowledge, describe what the line(s) of code does.

_____

```
1    Import java.util.Scanner;

2

3    Public class FirstProgram

4    {

5        public static void main(String[] args)

6        {

7                System.out.println("Hello out there.");

8                System.out.println("I will add two numbers for you.");

9                System.out.println("Enter two whole numbers on a line:");

10

11               int n1, n2;

12

13               Scanner keyboard = new Scanner(System.in);

14               n1 = keyboard.nextInt();

15               n2 = keyboard.nextInt();

16               System.out.println("The sum of those two numbers is ");

17               System.out.println(n1 + n2);

18       }

19   }
```

Using BlueJ

For Java, we will be using a new compiler called BlueJ. It is very similar to the Visual Studio Compiler that you used for C++. Here I would like to walk you through the process of coding, compiling, and running a program in BlueJ.

1) Lucky for us, BlueJ is a free, open-source program! It is available for both Windows and Mac OS. Begin by downloading and running the installer files from **www.bluej.org**
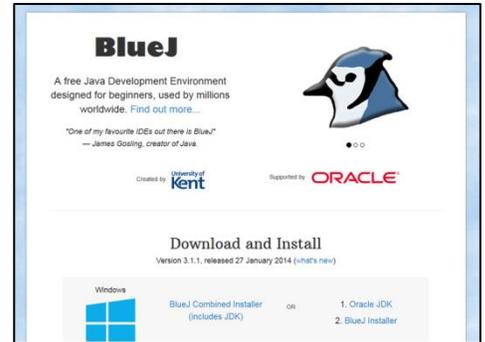
2) Run the program. Being by selecting **Project – New Project.** This will ask you to create a folder for your project. I highly suggest using your flash drive to save your work. This way you will have your files both at home and at school. Name this first folder **firstprogram.**
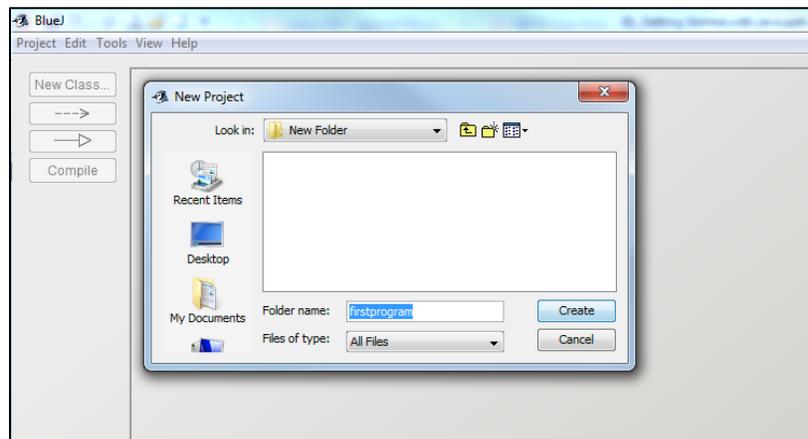
3) On the next screen, Select **New Class** from the left-hand menu. For most of our class we will be creating files of the Class Type **Class.** *(We will talk about Abstracts and Interfaces much later on in the course.)* Call this first class **FirstProgram. Click Ok**

Note: Typical naming convention calls for the name of classes to be one word and begin with an uppercase letter. If you have a name of a class that is made up of multiple words, capitalize the first letter of each new word. Class names cannot begin with a number or symbol.
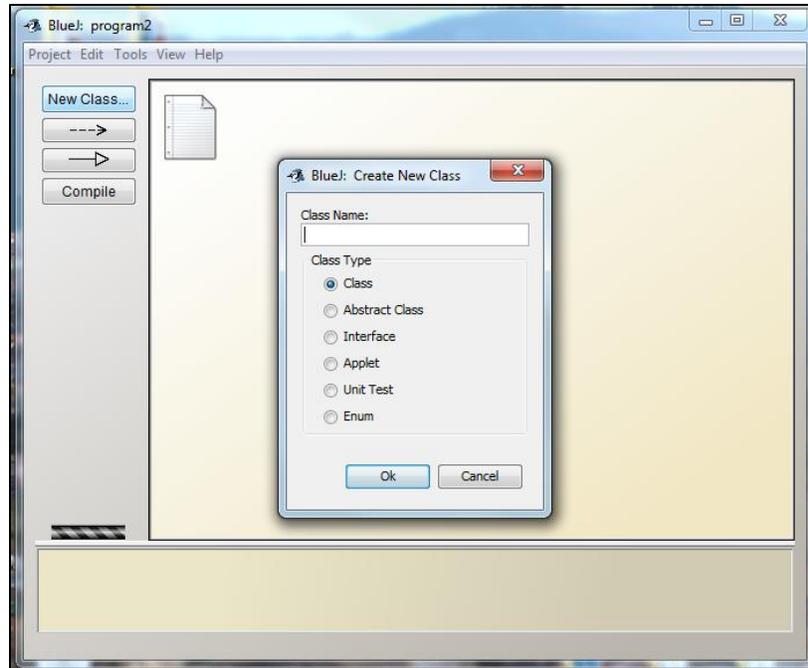
Valid Class Names	Invalid Class Names
FirstProgram	first program
StringManipulation4	String Manipulation 4
FinalExamAverages	2015FinalExamAvg

4) You should now see a yellow box with the class name in the BlueJ window. You can double click the class to begin programming. BlueJ will always give you a basic shell template of code to work with.
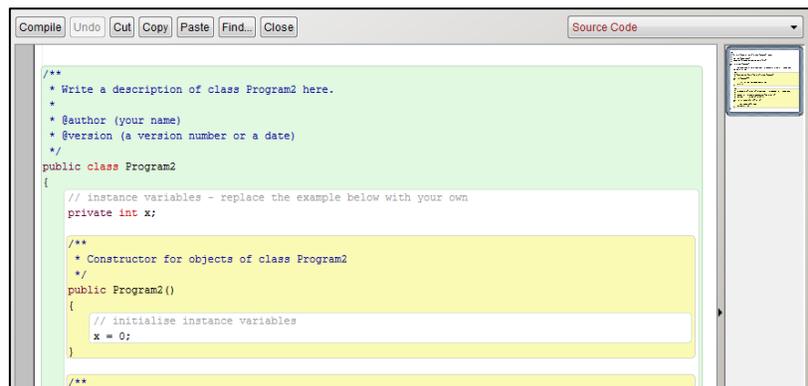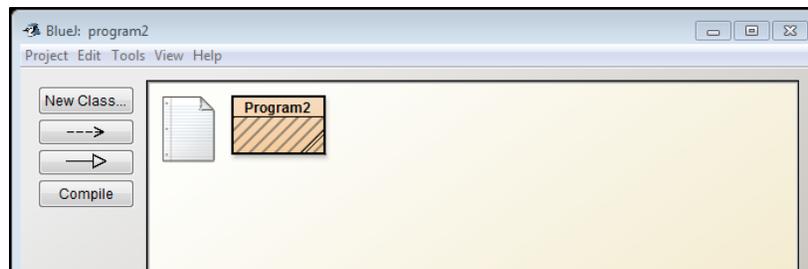
5) For our purposes, we are going to compile and run the sample program from page 3. Replace the entire Java template provided by BlueJ with the code below. You can copy and paste the following:

```
//AP Computer Science
//First Java Sample Program

import java.util.Scanner;

public class FirstProgram
{
    public static void main(String[] args)
    {
        System.out.println("Hello out there.");
        System.out.println("I will add two numbers for you.");
        System.out.println("Enter two whole numbers on a line:");

        int n1, n2

        Scanner keyboard = new Scanner(System.in);
        n1 = keyboard.nextInt();
        n2 = keyboard.nextInt();

        System.out.println("The sum of these two numbers is");
        System.out.println(n1 + n2);

    }
}
```
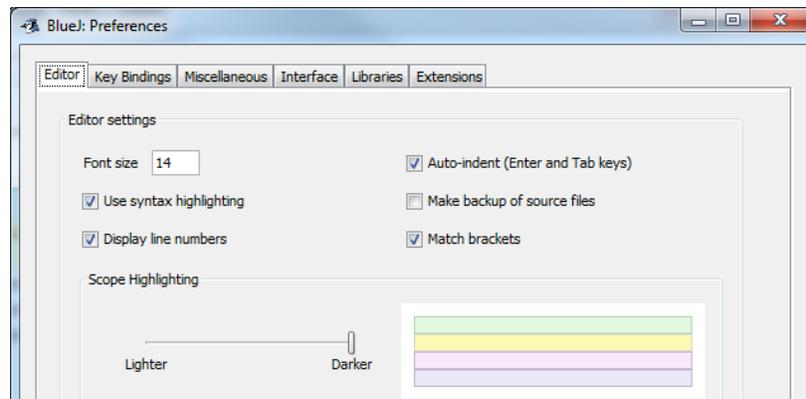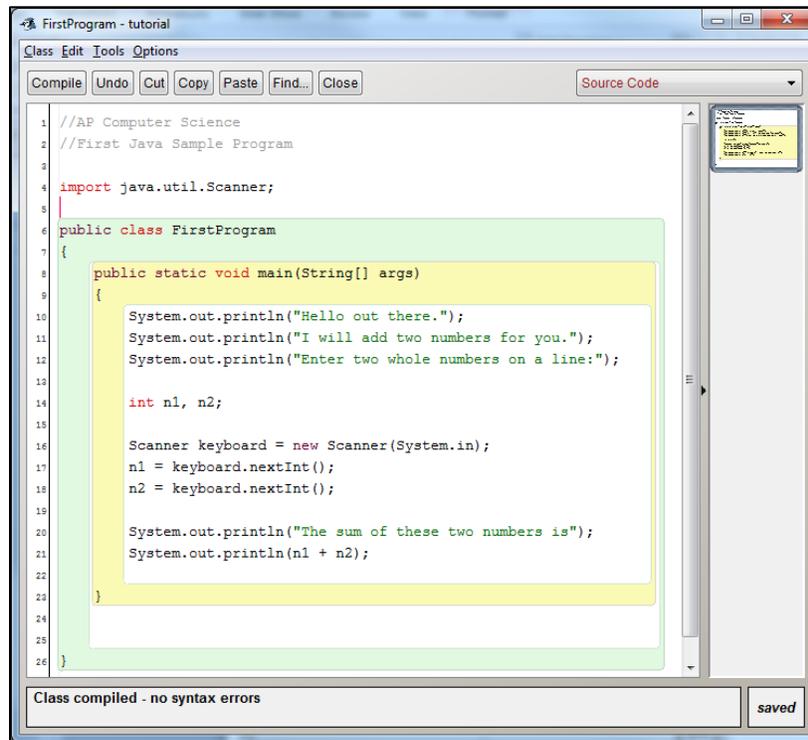
6) To make coding a bit easier to follow, BlueJ can mark your code with line numbers and highlighting. To set these preferences, select the **Options Menu – Preferences.** On the next window make sure **Use syntax highlighting** and **Display line numbers** are checked off. Click OK.

7) When you are done coding, **save your class** and click **Compile.** The compiler will now compile your code and do basic error checking. If there are errors, you will see them appear in the window at the bottom of screen. **In this example I have intentionally left out a semicolon at the end of line 14.** When you compile, you will see the line become highlighted, and the error message in the bottom window. If you add the semicolon (;) to the end of line 14 and recompile, you will see the error go away. The window should now say **Class compiled – no syntax errors.** Close the class.

```
//AP Computer Science
//First Java Sample Program

import java.util.Scanner;

public class FirstProgram
{
    public static void main(String[] args)
    {
        System.out.println("Hello out there.");
        System.out.println("I will add two numbers for you.");
        System.out.println("Enter two whole numbers on a line:");

        int n1, n2;

        Scanner keyboard = new Scanner(System.in);
        n1 = keyboard.nextInt();
        n2 = keyboard.nextInt();

        System.out.println("The sum of these two numbers is");
        System.out.println(n1 + n2);

    }

}
```

8) To run the compiled program, right click the class you just compiled and select the name of the main program you wrote. In this case select **void main (String[] args).** Then click ok on the next window.
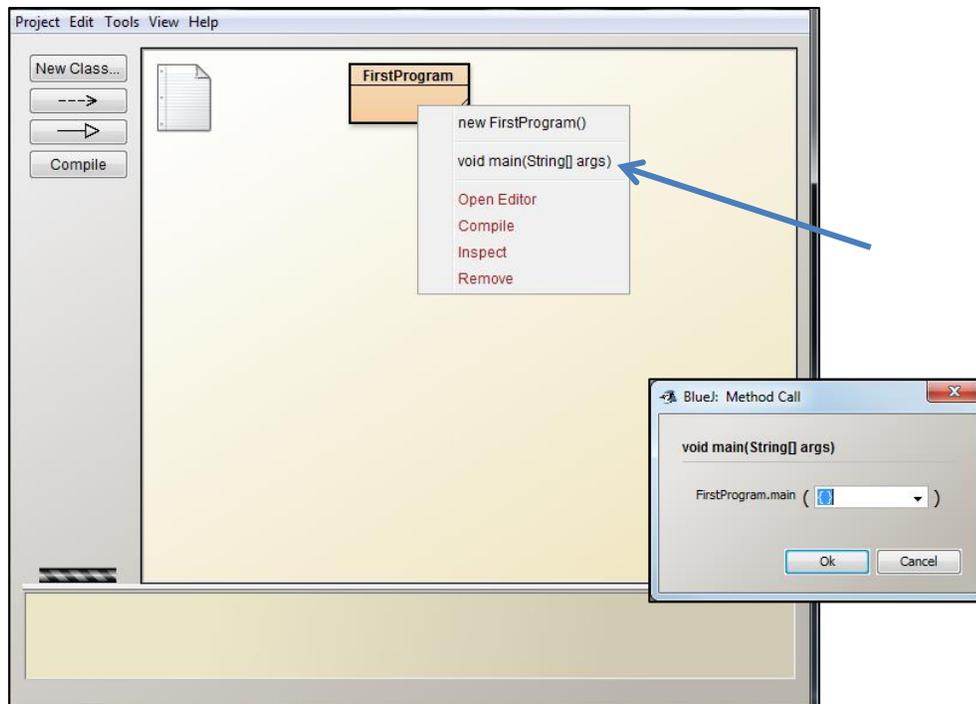
9) The terminal window will open, and you will see the output of the program.

10) Congratulations! You have written your first Java program!

Join our Google Classroom: Join Code **43huah**

Review of Programming Topics from C++

Much of the basic programming concepts that you learned in C++ are similar, if not identical, to Java. Let's see what you remember and show you how concepts can be a bit different in Java. For this summer, I will ask you to work through some lessons in a text book **Blue Pelican Java, by Charles Cook.** We will use this text book throughout the year in addition to another text, Java: An Introduction to Problem Solving, by Walter Savitch. You will pick up the Savitch text in September. I have the BPJ text in PDF format, which you can download from Google Classroom. It is under **Classwork – About/Resources**

Please read the following lessons and complete the questions and programs listed. Exercises responses should be typed. Projects should be written, compiled, and run using BlueJ.

As you are reading and doing these exercises, you may run into concepts, syntax, and/or vocabulary that you do not understand or are particularly challenging. Do your best to read/Google/YouTube these concepts on your own. If that isn't helpful, please keep a running list of questions you'd like to bring to the class. We can address these questions in September

Lesson	Page	Questions/Programs
1 – Hello World	Pg. 1-3	Project - From Me To You
2 – Variable Types	Pg. 2-3	Exercise on Lesson 2 #1-12
3 – Simple String Operations	Pg. 3-3	Exercise on Lesson 3 #1-12 Project – Name that Celebrity
4 – Using Numeric Variables	Pg. 4-3	Exercise on Lesson 4 #1-18 Project - Cheating on Your Arithmetic Assignment
5 – Mixed Data Types, Casting, and Constants	Pg. 5-2	Project – Mixed Results
7 – Input from the Keyboard	Pg. 7-3	Project – What's My Name?
8 – The Boolean Type and Boolean Operators	Pg. 8-3	Exercise on Lesson 8 #1-18
9 – The if Statement	Pg. 9-3	Exercise on Lesson 9 #1-20 Project – Even or Odd?
11 – The for-Loop	Pg. 11-4	Exercise on Lesson 11 #1-14
12 – The while & do-while loops	Pg. 12-3	Exercise on Lesson 12 #1-14 Programs 7 & 8 below

Program 7

Write a fragment of code that will compute the sum of the first n positive odd integers. Prompt the user to enter n for the number of digits they would like to compute. For example, if n is 5, you should compute 1 + 3 + 5 + 7 + 9.

Program 8

Write a program to read a list of nonnegative integers and to display the largest integer, the smallest integer, and the average of all the integers. The user indicates the end of the input by entering a negative sentinel value that is not used in finding the largest, smallest, and average values. The average should be a value of type double, so that it is computed with a fractional part.

Let me know if you have any questions. Feel free to email me at **apinnavaia@stedmundprep.org** if there is something that is really giving you a problem. Have a great summer and happy coding!