AP Computer Science
Unit 1. Writing Programs Using BlueJ

1. Open up BlueJ. Click on the Project menu and select "New Project." You should see the window on the right.

Navigate to wherever you plan to save your work. (In this example I saved to the desktop – don't you do that.)

Enter "Unit 1" where it says Folder name and then click the Create button.

2. After you've clicked Create button, the screen to the right should appear.

If you double-click on the file icon you'll see that it's a Read-Me file. This can be a useful way to document each project. Use it if you want but it is not a requirement.

3. Click on the New Class button and window appears. Enter "First" and click OK.
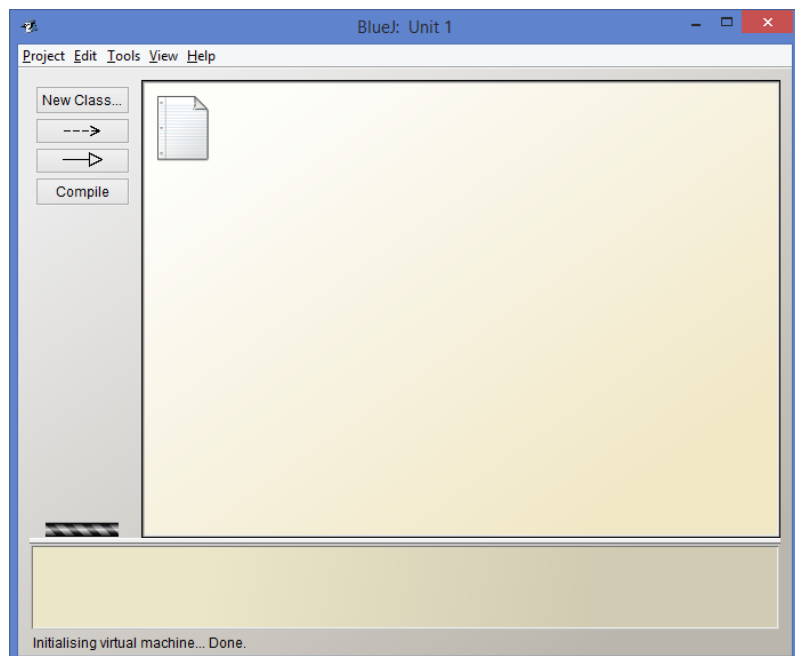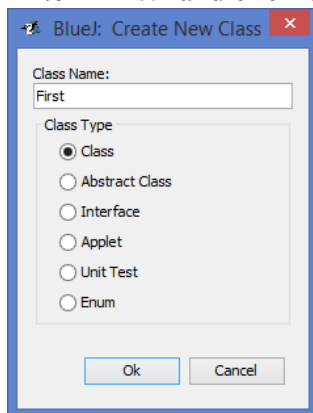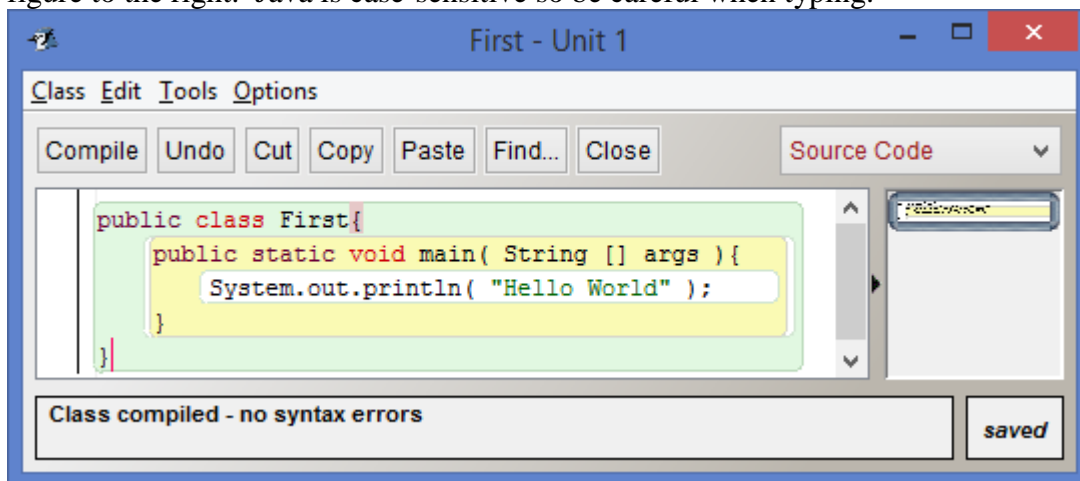
4. Your screen should look like this:

Double-click on the First icon.
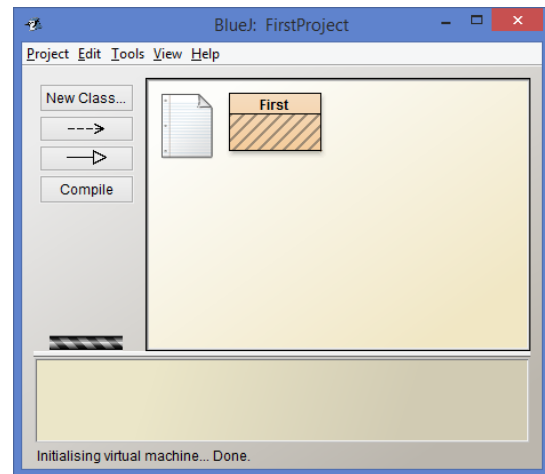
5. Delete everything in the First file. Then type in the following code so that it looks like the figure to the right. Java is case-sensitive so be careful when typing.


```java
public class First{
    public static void main( String [] args ){
        System.out.println( "Hello World" );
    }
}
```

6. Click the Compile button. If the message "Class compiled – no syntax errors" appears (as shown above, then Great, you got it right. All java programs must successfully compile before they can be run. If there were errors, fix them.

7. Now go back to the earlier window and right-click on the First icon and select "void main(String[] args)". The window below will appear.

BlueJ: Unit 1

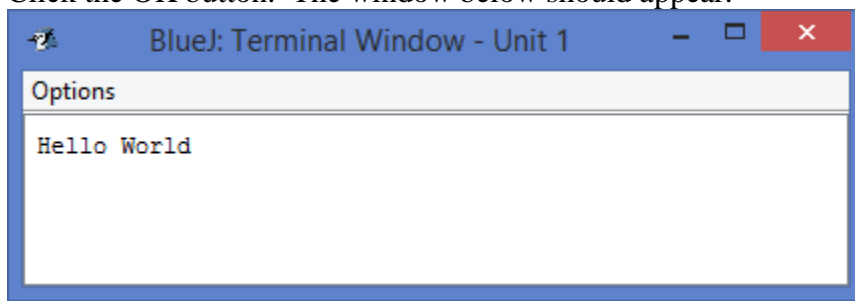Project Edit Tools View Help

New Class...

Compile

First

new First()

void main(String[] args)

Open Editor
Compile
Inspect
Remove

BlueJ: Method Call

void main(String[] args)

First.main ({})

Ok   Cancel
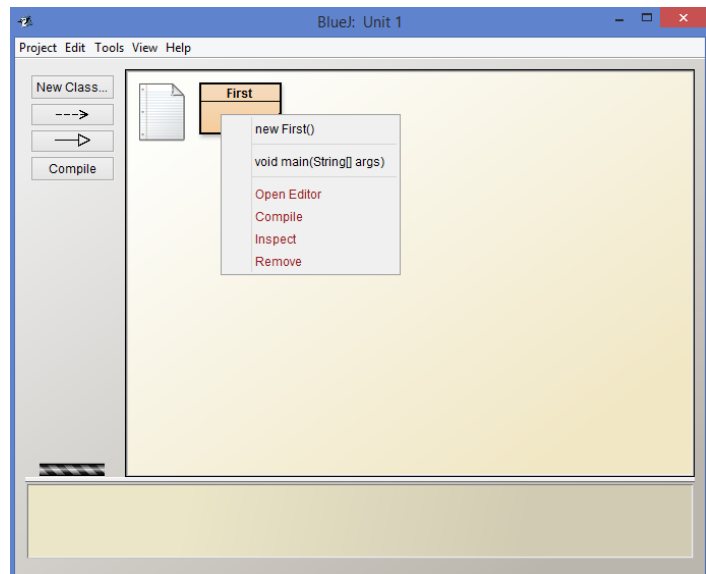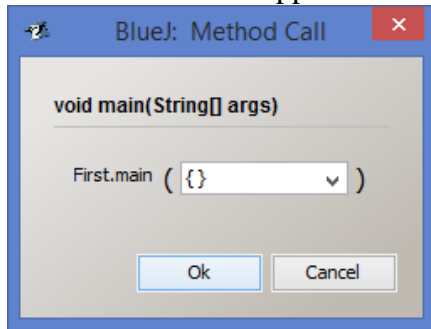
Do not enter anything into the textbox.
Click the OK button. The window below should appear.

BlueJ: Terminal Window - Unit 1

Options

Hello World
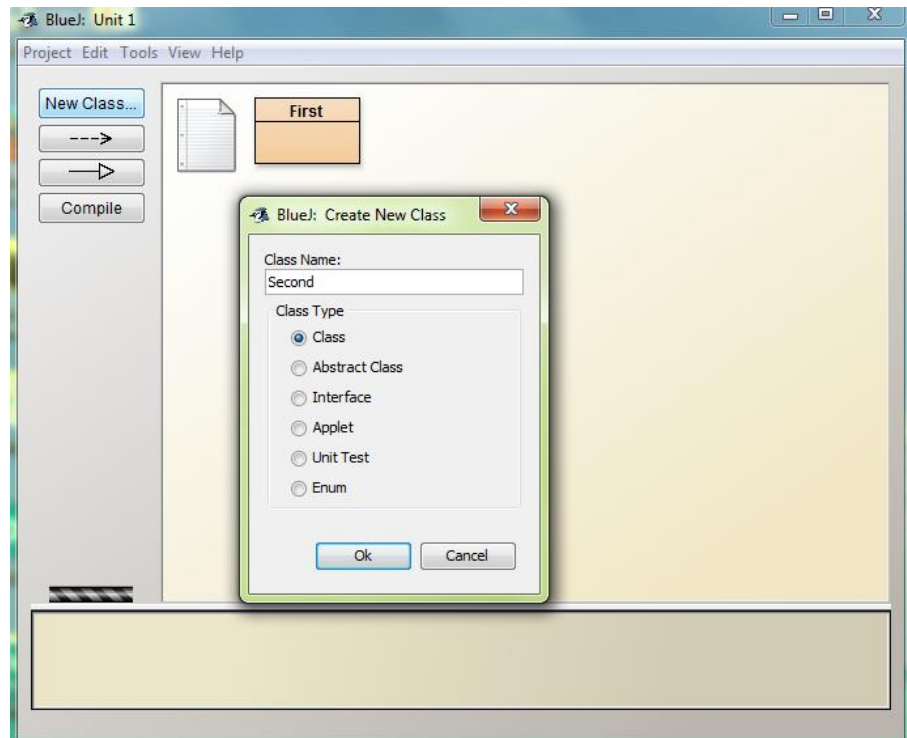
public class First {

```
    public static void main(String[] args) {

            System.out.println( "Hello World" );

    }
```

}

**Second Program**. Click on the New Class button. Name the class Second and click OK.

BlueJ: Unit 1
Project Edit Tools View Help

New Class...
--->
Compile

First

BlueJ: Create New Class

Class Name:
Second

Class Type
- Class
- Abstract Class
- Interface
- Applet
- Unit Test
- Enum

Ok   Cancel

Double-click on the icon labelled Second. Delete all the existing code and replace it with the following.

Second - Unit 1
Class Edit Tools Options

Compile Undo Cut Copy Paste Find... Close   Source Co

```
public class Second{
    public static void main( String [] args ){
        int x = 5;
        System.out.println( "The value of x is " + x );
        x = x + 10;
        System.out.println( "The value of x is " + x );
    }
}
```

Class compiled - no syntax errors

Click Compile. If there are errors, fix them and click Compile again.

Notice that + has two different meanings. In the second line of the main method it *concatenates* the phrase in quotes with the value of the variable. In the third line + means addition.
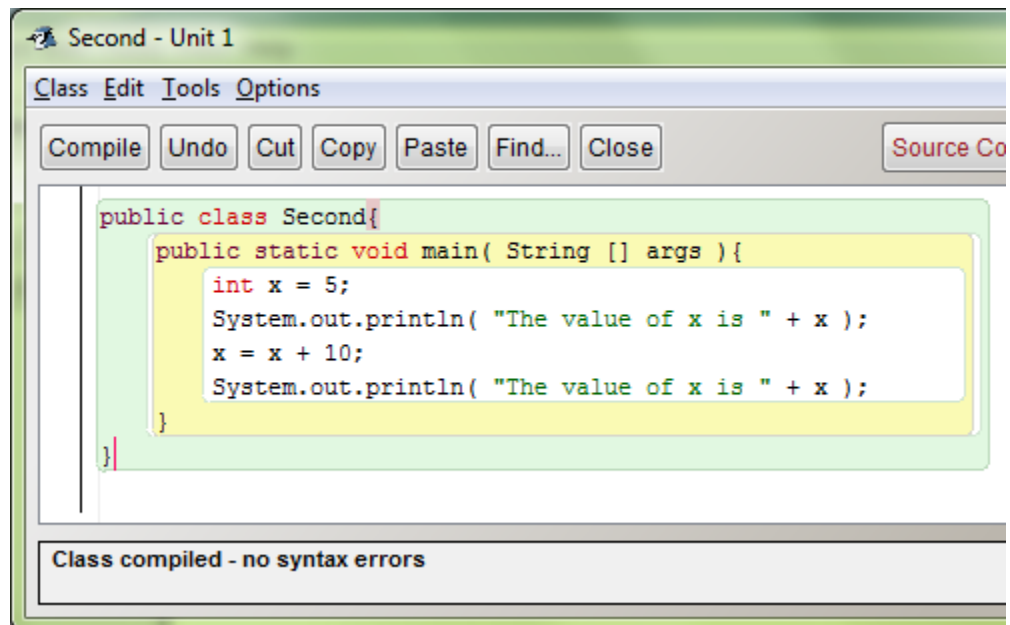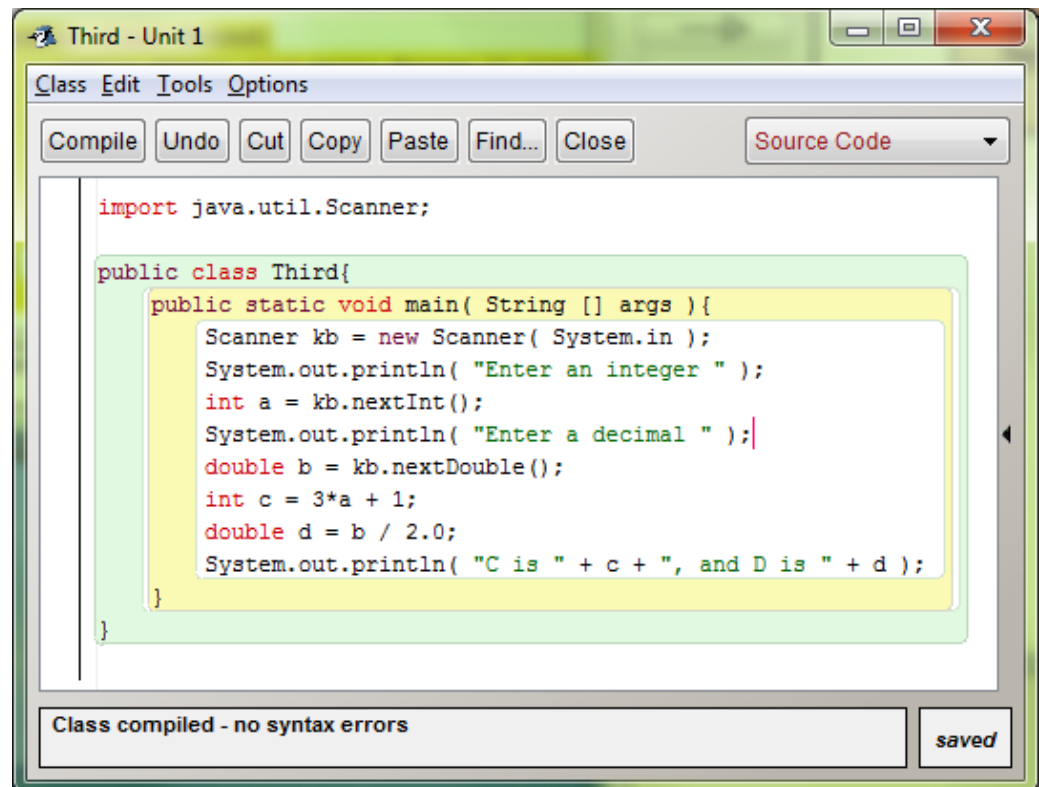
Go to the main window. Right-click on the icon labelled Second and select the main method. Click OK and you should see this Terminal window. You may also see "Hello World" if you ran the two programs back to back without clearing the Terminal window first.

BlueJ: Termi...
Options

The value of x is 5
The value of x is 15
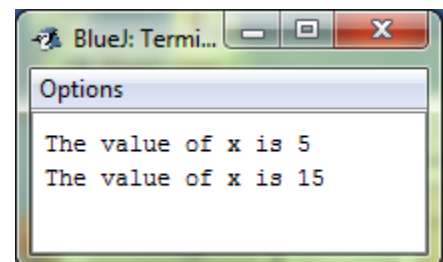
Third Program.
Create a class, named Third, and enter this code.

Compile it and run the program (by right-clicking the icon and calling the main method).

Third - Unit 1

Class Edit Tools Options

Compile | Undo | Cut | Copy | Paste | Find... | Close   Source Code
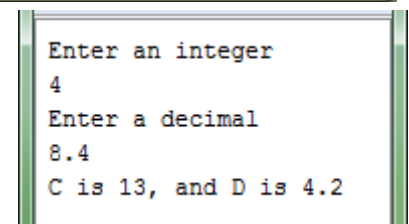
```java
import java.util.Scanner;

public class Third{
    public static void main( String [] args ){
        Scanner kb = new Scanner( System.in );
        System.out.println( "Enter an integer " );
        int a = kb.nextInt();
        System.out.println( "Enter a decimal " );
        double b = kb.nextDouble();
        int c = 3*a + 1;
        double d = b / 2.0;
        System.out.println( "C is " + c + ", and D is " + d );
    }
}
```

Class compiled - no syntax errors   saved

If you enter 4 for the integer and 8.4 for the decimal, your Terminal window should look like this.

```
Enter an integer
4
Enter a decimal
8.4
C is 13, and D is 4.2
```

This program shows us how to create a Scanner object. A Scanner object has methods that allow us to read information into our program. We will cover objects and methods in great detail starting in Unit 4. Here's what you need to know for now.

1) If you want the user to enter information into your program, the first line of your program will be:
    import java.util.Scanner;
This tells the compiler that the program needs to know about the Scanner class and that information can be found in a package named java.util.

2) To create a Scanner object you will use a statement like this:
    Scanner *variable_name* = new Scanner(System.in);
The variable name must follow the same rules as we discussed on the first page of the notes (i.e. start with a letter, etc.)

3) If you want the user to enter an integer, you must call the nextInt method. If you want the user to enter a decimal, you must call the nextDouble method.

4) When you call nextInt or nextDouble, you must use the variable name of the Scanner object and the method will be on the right side of an assignment statement. And don't forget the parentheses – that's a signal to the compiler that nextInt and nextDouble are methods.

Fourth Program. In this program the user will enter the length and width of a rectangle and the program will calculate and display the area and perimeter of the rectangle. Copy the outline below, fill in the blanks, and get the program to run.

_____

```
public class Rectangle{
        public static void main( String [] args ){
                Scanner sam = new Scanner( System.in );
                System.out.println( "Enter the length " );
                int length = sam._____
                System.out.println( "Enter the width " );
                int width = _____.nextInt();
                int area = _____
                int perimeter = _____
                System.out.println( "The area is " + area );
                System.out.println( "The perimeter is " + perimeter );
        }
}
```

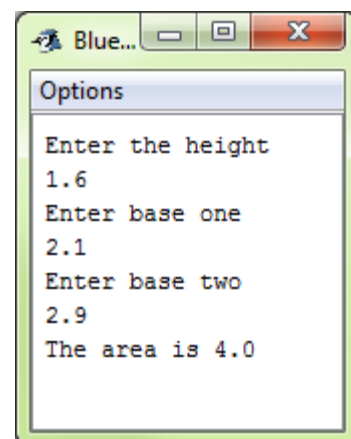Here's what your program should display if the user enters 5 and 11.

```
Blue...                          □  X

Options

Enter the length
5
Enter the width
11
The area is 55
The perimeter is 32
```
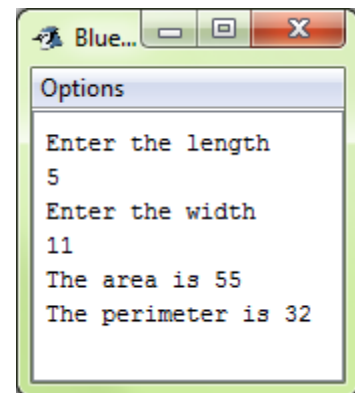
Fifth Program. This program will implement the formula for calculating the area of a trapezoid:

$$area = height * \frac{(base_1 + base_2)}{2}$$

Write a program where the user enters the height and two bases as decimals and the program calculates and displays the area.

Here's what your program should display if the user enters 1.6 for the height, 2.1 for $base_1$ and 2.9 for $base_2$.

```
Blue...                    □  X

Options

Enter the height
1.6
Enter base one
2.1
Enter base two
2.9
The area is 4.0
```

When should we declare a variable as an int and when should we declare it as a double?
Programs run faster and more accurately with ints. So wherever possible, it is better to use an int. However, there are circumstances where you must use a double. In this class I will usually tell you which data type to use. If I don't, then use common sense. If the variable represents the number of people in a room – use an int. If the variable represents an interest rate – use a double. If it is ambiguous then use your best judgment.

In the above paragraph I said that programs run more accurately with ints. Here's why. All data are represented as a series of ones and zeros (binary form). Certain decimal values cannot be represented exactly in binary form and so small "floating point" errors can appear. These types of errors cannot appear when working with ints. We will not concern ourselves with these errors but you should be aware that they can occur.

Here's an example. The output is to the right. The third line, 12.9 is correct but the 7.5 – 2.1 equals 2.1 so the last line is off by 0.0000000000000004. This is obviously a very tiny error but you should be aware that they can occur.

```
public class Floating_Point_Error{
        public static void main( String [] args ){
                double x = 7.5;
                double y = 5.4;
                System.out.println( x );
                System.out.println( y );
                double z = x + y;
                System.out.println( z );
                z = x - y;
                System.out.println( z );
        }
}
```



```
Options

7.5
5.4
12.9
2.0999999999999996
```
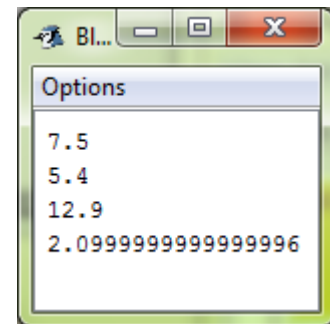
Sixth Program. This program should convert degrees Fahrenheit to degrees Celsius. The formula is:

$$C = \left(\frac{5}{9}\right)(F-32)$$

Fahrenheit should be an integer value but the result is a decimal. And just in case you need it, here are some test data:

0 Fahrenheit = -17.7777777778 Celsius
32 Fahrenheit = 0 Celsius
212 Fahrenheit = 100 Celsius

Seventh Program. Write a program where the user enters a positive integer that represents a certain number of hours. The program then displays a message that displays the number in terms of days and hours. For example: if the user enters 51, then the program displays:

        51 hours = 2 days and 3 hours

If the user enters 18, the program displays:

        18 hours = 0 days and 18 hours.

        You will need to use the mode operator.

Eighth Program. One flower costs $7 but a dozen flowers cost "only" $70. Write a program where the user enters the number of flowers and the program displays the cost. For example, if the user enters 27 then the program displays:

        27 flowers will cost $161

If the user enters 11 then the program displays:

        11 flowers will cost $77

Ninth Program. Write a program that runs like this - the user enters a number, let's say *10*, and the program displays:

        Half of 10 is 5
        Half of 5 is 2.5
        Half of 2.5 is 1.25
        Half of 1.25 is 0.625