

# Advanced Placement Computer Science

## Unit 1 – An Introduction to Hardware, Software, and the Internet

## Unit 2 – Your First Java Programs

Each unit, you will have an assignment that includes some reading, some questions from the reading, called Review Exercises, and between one and three programming exercises.

### Calendar of Meeting Places for August

				16	17 <u>Class</u> Intro to Course	18
19	20 <u>Class</u> Lesson 1.1	21 <u>Class</u> Lesson 1.2, Work on Assignment, and Introduce Long Term Project (Blackjack)	22	23 <u>Class/Lab</u> Lesson 2.1 & 1 <sup>st</sup> Programs	24	25
26	27 <u>Class</u> Lesson 2.2	28 <u>Lab</u> 1 <sup>st</sup> Programs	29	30 <u>Class</u> Begin Unit 3 & Introduce “Close Reading” <b>(Quiz today)</b>	31	

Assignment Type	Description
Reading	Java Methods – Chapter 1, Chapter 2 Note: Chapter 1 is online at: <a href="http://www.skylit.com/javamethods">http://www.skylit.com/javamethods</a>
Homework Exercises <b>(Due Thursday, August 30)</b>	<u>Questions</u> : Complete Handouts  <u>Programming</u> None

# Your First Java Programs

(keep this for reference!!)

1. Turn on your computer. All of your work will be saved on your USB Flash Drive.
2. Before we do anything, let's try to be organized this year. We are going to try to use "the cloud" this year in all that we work on. If you do all of your work on the Desktop, at the end of each day you simply drag your folder to your Google Drive. The next day in class, drag the entire folder back the Desktop and you are good to go! For this unit, create a folder called **Unit 2** (on the Desktop). Within the Unit 2 folder, create another folder called **HelloWorld**.
3. On the Desktop, click on **IntelliJ IDEA Community Edition** to open our tool that we will use.
4. You are now using **IntelliJ**. This is called an Integrated Development Environment (IDE) because you can type in programs, build them so that the program will be converted to binary, run your program, and in case it does not work, debug your program (fix errors).
5. You now want to create a new project. Click on **+Create New Project**. Select **Next** twice. Here you need to give your project a name. Easiest thing to do is to browse for your folder, **HelloWorld**, that you created in step 2. You can name it anything you like, but something that is descriptive is better, like **HelloWorld**. Select that folder and your project will be named the same as the folder.
6. You now want to create a new Java file. Click to expand your project then right click on the src folder. Select **New| Java Class** and name it **HelloWorld (no space between the words)** press OK.
7. You may now type in the program on page 23 of your textbook.
8. To run your program, you first have to **Build** it. Expand the **src** folder and right click **HelloWorld**. Scroll down to select **Build Module 'HelloWorld'**. If there are no errors, right click **HelloWorld**. Scroll down to select **Run 'HelloWorld.main()'**.

After you have successfully executed the program, describe below what happens.

# Your First Java Programs – Exploring with Turtle Objects

1. Create a new project called **ExploreWithTurtle**.
2. Create a new java class called **ExploreWithTurtle**.
3. Type in the program as shown below.

```
// Name: Your name here
// Last Changed: The last day that you made changes

import java.awt.Color;

/**
 * This class is our first attempt at programming. In this area, you should include
 * details as to what this program does.
 */
public class ExploreWithTurtle
{
    public static void main (String [] args)
    {
        Turtle sam;
        sam = new Turtle();
        sam.paint (90, 12);
        sam.move (-180, 6);
        sam.paint (90, 6);
        sam.move (90, -6);
        sam.paint (0, 12);
        sam.move (150, 6);
        sam.swingAround (25);
    }
}
```

4. You need one more file in your project, *Turtle.java*. You will find this on my web site on the Assignments Link. Please download and save to your **src** folder for **ExploreWithTurtle** project
5. **Build and Run your Program like you did in HelloWorld.**

## Challenge (*if you are shooting for an A in this class*)

Write an application program that creates one `Turtle` and has it draw a house 200 pixels wide and 150 pixels tall, with a door and four windows. This house should have a 45 degree roof pitch and contain a chimney.

# Methods of the Turtle Class

## About the nine Turtle methods:

- `new Turtle()` creates a Turtle object in the center of a drawing surface 760 pixels wide and 600 tall. The Turtle initially faces east and carries a black paintbrush. In the following, `sam` is a Turtle variable to which you have assigned a Turtle object.
- `sam.paint(angle, dist)` tells `sam` to turn counterclockwise by `angle` degrees and then go forward by `dist` pixels, leaving a trail of the current drawing color.
- `sam.move(angle, dist)` tells `sam` to turn counterclockwise by `angle` degrees and then go forward by `dist` pixels, without leaving any marks.
- `sam.swingAround(dist)` tells `sam` to draw a circle of radius `dist` pixels with `sam` at the center.
- `sam.fillCircle(dist)` tells `sam` to draw a circle of radius `dist` pixels with `sam` at the center, and fill its interior with the current drawing color.
- `sam.fillBox(width, height)` tells `sam` to draw a rectangle `width` pixels wide and `height` pixels tall, with `sam` at the center, and fill its interior with the current drawing color.
- `sam.switchTo(col)` tells `sam` to change the current drawing color to `col`, which can be any of BLACK, GRAY, BLUE, GREEN, RED, YELLOW, ORANGE, PINK, MAGENTA, and WHITE. Put "Color." in front of the name of a color you use in a program, unless you use it in a subclass of the Turtle class.
- `sam.say("whatever")` tells `sam` to print whatever is within the quotes.
- `sam.sleep(milli)` tells `sam` to suspend action for `milli` milliseconds.