Summer Work Packet for AP Computer Science A
2023-2024

Welcome to AP Computer Science A!

Please complete the following packet. It will be due Tuesday after Labor Day weekend and count as a quiz grade.

Since AP Computer Science A is strictly about Java programming, you will need to use a Java compiler.

You can either download free software from Blue J, Netbeans, or Eclipse or you can use a free online browser such as:
https://www.onlinegdb.com/online_java_compiler

Using the videos HelloWorld, this tutorial on variables, and the following NOTES, complete the project at the end of the packet and print out your program, AND answer exercises 1-12. This will allow us to go through the introductory parts of the curriculum quickly and spend more time on the more difficult concepts that come up later in the year.

AP Computer Science A will help give you a strong foundation in computer programming, particularly in the Java language. It's going to be a great year!

Lesson 1…..Hello World

Program Skeleton:

Enter the following program skeleton, compile (prepare it to run), and then run (execute). Your instructor may have you give it a specific project name; otherwise, call the project *Lesson1*.

> **If you do not know how to enter and execute a program, ask your instructor, or use the appendices in this book for two of the more popular programming environments. See <u>Appendix N</u> for the BlueJ environment and <u>Appendix O</u> for the JCreator environment.**

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

        }
}
```

At this point don't worry about what any of this means. It's just something we must do every time. Soon we will learn the meaning of all of this. For now it's just the skeleton that we need for a program.

Adding some meaningful code:

Now, let's add some meaningful code inside the *main* method. (Notice this word, **method**. We will constantly refer to **methods** throughout this course.) We will also add a **remark**.

```
public class Tester //We could put any name here besides Tester
{
        public static void main(String args[])
        {
                System.out.println("Hello world");
        }
}
```

Remarks:

Notice the rem (remark) above that starts with //. You can put remarks anywhere in the program without it affecting program operation. Remarks are also called comments or notes.

Printing:

System.out.println("Hello world"); is how we get the computer to printout something. Notice the trailing semicolon. Most lines of code are required to end in a semicolon.

Now try putting in some other things in the *println* parenthesis above. Each time recompile and run the program:

1. "Peter Piper picked a peck of pickled peppers."
2. "I like computer science."

3. 25/5
4. 4 / 7.0445902
5. 13 * 159.56

Two *println*s for the price of one:
Next, modify your program so that the *main* method looks as follows:

```
public static void main(String args[])
{
        System.out.println("Hello world");
        System.out.println("Hello again");
}
```

Run this and note that it prints :
```
Hello world
Hello again
```

Printing "Sideways":
Now remove the *ln* from the first *println* as follows:
```
public static void main(String args[])
{
        System.out.print("Hello world");
        System.out.println("Hello again");
}
```

Run this and note that it prints:
```
Hello worldHello again
```

An in-depth look at rems:
Let's take a further look at rems. Consider the following program (class) in which we wish to document ourselves as the programmer, the date of creation, and our school:

```
public class Tester
{
        //Programmer: Kosmo Kramer
        //Date created: Sept 34, 1492
        //School: Charles Manson High School; Berkley, Ca

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

Block rems:

It can get a little tedious putting the double slash rem-indicator in front of each line, especially if we have quite a few remark lines. In this case we can "block rem" all the comment lines as follows:

```
public class Tester
{
        /*Programmer: Kosmo Kramer
        Date created: Sept 34, 1492
        School: Charles Manson Junior High; Berkley, Ca*/

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

Notice we use /* to indicate the start of the block and */ for the end. **Everything** between these two symbols is considered to be a remark and will be ignored by the computer when compiling and running.

Lesson 2…..Variable Types (*String*, *int*, *double*)

Three variable types:

(A good way to learn the following points is to modify the code of the "Hello World" program according to the suggestions below.)

1. *String*….used to store things in quotes….like "Hello world"
   Sample code:
```
public static void main(String args[])
{
        String s = "Hello cruel world";
        System.out.println(s);
}
```

2. *int* ….used to store integers (positive or negative)
   Sample code:
```
public static void main(String args[])
{
        int age = 59;
        System.out.println(age);
}
```

3. ***double*** ....used to store "floating point" numbers (decimal fractions). *double* means "double precision".

      Sample code:

```
public static void main(String args[])
{
        double d = -137.8036;
        System.out.println(d);

        d = 1.45667E23;  //Scientific notation…means 1.45667 X 10^23
}
```

Declaring and initializing:

When we say something like

    double x = 1.6;

we are really doing **two** things at once. We are **declaring** *x* to be of type *double* **and** we are **initializing** *x* to the value of 1.6. All this can also be done in **two** lines of code (as shown below) instead of one if desired:

    double x; //this declares x to be of type double
    x = 1.6; //this initializes x to a value of 1.6

What's legal and what's not:

    int arws = 47.4; //illegal, won't compile since a decimal number cannot "fit" into an
                 //integer variable.
    double d = 103; //legal…same as saying the decimal number 103.0

Rules for variable names:

Variable names must begin with a letter (or an underscore character) and cannot contain spaces. The only "punctuation" character permissible inside the name is the underscore ("_"). Variable names cannot be one of the reserved words (key words…see <u>Appendix A</u>) that are part of the Java language.

Legal names	Illegal names
Agro	139
D	139Abc
d31	fast One
hoppergee	class
hopper_gee	slow.Sally
largeArea	double
goldNugget	gold;Nugget
	hopper-gee

Variable naming conventions:

It is traditional (although not a hard and fast rule) for variable names to start with a lower case letter. If a variable name consists of multiple words, combine them in one of two ways:

    bigValue… jam everything together. First word begins with a small letter and subsequent words begin with a capital.

    big_value… separate words with an underscore.

Complete the following Project in the compiler of your choice and then copy to a Google document and print out to turn in.

Project... From Me To You

Create a new project called *FromMeToYou* having a *Tester* class with the following content. Also include remarks above *public class Tester* that identifies you as the author along with the date of creation of this program:

```
//Author: Charles Cook
//Date created: Mar 22, 2005
public class Tester
{
   public static void main(String args[])
   {
        ...
   }
}
```

Supply code in the place of ... that will produce the following printout:

```
From: Bill Smith
Address: Dell Computer, Bldg 13
Date: April 12, 2005

To: Jack Jones

Message: Help! I'm trapped inside a computer!
```

Answer the following question to the best of your ability using the notes and the video provided.

Exercise on Lesson 2

1. What are the three main types of variables used in Java and what are they used to store?

2. What type of variable would you use to store your name?

3. What type of variable would you use to store the square root of 2?

4. What type of variable would you use to store your age?

5. Write a single line of code that will create a double precision variable called *p* and store 1.921 X 10^{-16} in it.

6. Write a single line of code that will create an integer variable called *i* and store 407 in it.

7. Write a single line of code that will create a *String* variable called *my_name* and store your name in it.

8. Write a line of code that will **declare** the variable *count* to be of type *int*. Don't initialize.

9. Write a line of code that **initializes** the double precision variable *bankBalance* to 136.05. Assume this variable has already been declared.

10. Which of the following are legal variable names?
    scooter13 139_scooter homer-5 ;mary public doubled double ab c

11. Which of the following is the most acceptable way of naming a variable. Multiple answers are possible.
    a. GroovyDude
    b. GROOVYDUDE
    c. groovyDude
    d. Groovydude
    e. groovy_dude
    f. groovydude

12. Comment on the legality of the following two lines of code.
    double dist = 1003;
    int alt = 1493.86;