Mr. Rogers' Syllabus: AP Computer Science A (Part I of a 2 Year AP Computer Science Program)

What to Expect:

Computer science is one of the most dynamic job markets in America. It has surpassed engineering in the number of jobs and rivals it for pay. Within the computer science job market, Software Engineer is often listed as one of the best jobs in America from the standpoint of job satisfaction, work environment, and pay.

At Mr. Rogers high, AP Computer Science A is the first year of a two year AP Computer Science Program. The course goes way beyond learning how to use applications like Word, Excel and PowerPoint. It focuses on using the Java language to design and build applications. It's a college level course and at the end of the first year, students will be required to take the AP Computer Science A exam. If they pass they may receive college credit for one semester of computer science.

After completing the AP Computer Science A course, students may elect to continue to the second year of AP Computer Science. At the end of the second year, students are required to take the AP Computer Science AB exam. If they pass they may receive college credit for an additional semester of computer science.

AP Standards:

- **Topic Outline:** This link give the official AP computer science topic outline.
- **Java Subset:** The Java language is huge and so AP computer science will focus on a limited but very powerful part of the subset.
- **Case Study**: A case study is included in the requirements. Students will learn how to modify its functionality.

Computer Lab: Our computer lab contains up-to-date Pentium IV computers with 2 Gig of RAM and 19" flat screen monitors as well as a promethean board for displaying information to the entire class. All classes (5 days a week) will be held in the lab.

Who Should Take this Class: Students with an interest careers related to: engineering, the computer professions, bioinformatics, genetics, physics, chemistry, math, or information management of any type.

Credit: One unit of math. Meets the 1 year of computer science requirement for graduation.

Prerequisites: Honors Algebra II and and programming experience or permission of the instructor. **NOTE: This is a College level class NOT a beginner's or an applications class (such as word processors, spread sheets, etc.)** Students signing up for this class should be familiar with basic computer terms and have some programming experience before they start. A strong math background with a PSAT math score of 58 or higher is a positive indication of readiness for the class.

Grading and Assignments

Grading: (For details see Mr. Rogers's Syllabus - Information Common to All Classes.) Tests will be the single largest item and be written as close to AP exam standards as possible. All tests will be pencil and paper just like the AP Test.

Course work will generally be finished by the end of the third quarter leaving the rest of the year for review and completion of the major project (see below). The first semester exam will be taken from AP test type materials. The exam grade will give you an indication of your standing on the future AP exam in enough time to take corrective action if needed.

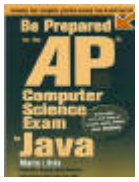
The fourth quarter grade will consist primarily of

Computer Lab Hours: The lab will be open on Mr. Rogers' lunch shift and after school for 90 min. every day except Wed.

Resources and Materials for Class

TextBooks : Mr. Rogers will provide the following books:

Java Methods A & AB Object-Oriented Programming and Data Structures AP Edition, Maria Litvin, Gary Litvin, 2006, by Skylight Publishing,
ISBN 978-0-9727055-7-8



Be Prepared for the AP Computer Science Exam in Java (Paperback)
by Maria Litvin

Addison-Wesley's Review for the Computer Science AP Exam in Java (Paperback)
by Susan Horwitz

Web Page Resources
All of the following are maintained by Mr. Rogers and can be reached via links from his teacher homepage.

1. **Objectives Provided by Mr. Rogers:** The daily objectives used in class along with programming and homework assignments are all available online.
2. **Textbook Page**: this page has lots of Java information including student files which you will want to download.
3. **Java Applications Programming Interfaces (API)** this is a primary source for information about the Java language.
4. **Google:** googling just about any Java topic will bring up tutorials or other useful information.

Materials for Class

practice AP test grades and the major project. Generally, there is a high correlation between the practice exams and your grade on the AP test. **It will be nearly impossible to get an "A" in the 4th quarter and not pass the AP Test.**

Regular Programming Assignments: every new unit will have moderate sized programming assignments which are due before the unit test. These assignments are listed in the on-line syllabus. Time will be allotted in class to work on these programs but it will normally also require at least some work outside of class.

Homework Assignments: these are listed in the on-line syllabus and are to be written out and stapled to the back of the unit test when it's turned in. No late homework will be accepted.

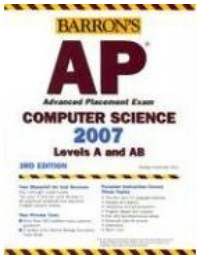
Class Work: We will frequently have brief programming assignments in class to illustrate a specific objective. These will normally be due no later than the next class session.

Personal Software Project: starting with milestones in December and ending with a completed project in May, each student will be required to design and create a Personal Software Project which answers a research question or meets a defined purpose such as a game or educational program.

The project gives students the opportunity to put to use everything they've learned. Past students have said that they learned more from the personal project than any other part of the course.

Click **here** for a detailed description of the project. **Be aware that the project will seriously impact both the 3rd and 4th quarters. It will be nearly impossible to get an "A" for the year without completing the project.**

1. **A USB thumb drive** for backing up programs. Note: **you are responsible for backing-up your programs**. Don't assume they will remain on the hard drive of your computer.
2. Students are not required to have a computer with an internet connection at home but it's recommended.
3. Students should download BlueJ (here) to use at home. This is the software used for programming in class.
4. **All students should purchase the Barron's Ap Computer Science Study Guide*** - How to Prepare for the AP Computer Science Exam by Roselyn Teukolsky.



Sources of Help: Student tutors will typically be available at lunch time and at selected times after school.

Mr. Rogers AP Computer Science A - First Quarter Objectives

Essential Question: How do you write a program?

Chapter 1 & 2: Software Basics

(II Program Implementation, III Program Analysis, VI Computing in Context)

First Day

Introduction - Hand out syllabus

1. Use console output (System.out.println)

Programming Assignment 1: Write a program that outputs "Hello World" to the screen.

2. Define the term debugging.
3. Define compiling error and describe a way to cause one.

Deliberately delete a ";" from the above program and note the error message.
Deliberately alter other elements of the program to see what happens

Second Day

4. Describe how byte code and the Java virtual machine work together.
5. Give examples of different operating systems and explain why the Java virtual machine is needed.
6. Explain what compilers and interpreters do with respect to Java.
7. Define encapsulation and information hiding--example: TV, System.out.println

Programming Assignment 2: Write a program that inputs (your name) and (your age) on the command line and outputs "Hello (your name) you are (your age) years old".

8. Define run-time error and describe a way to cause one. (Usually throws an exception.)

Deliberately cause a run time error in the above program and note what happens.

9. Define the term iteration and describe a way to do it--example: pounding a nail, "for" loop

Programming Assignment 3: Modify Programming Assignment 2 with a "for loop" so that the message repeats itself 10 times. once the program is running, modify it so that it has an infinite loop and see what happens when you run it.

10. Define logic error and describe a way to cause one.
11. Be aware that while a program can branch to a different part of the code, it otherwise executes sequentially from the top to the bottom for a given method.
12. Describe one of the key strategies in the creation of complex software: functional decomposition (breaking down an overwhelmingly complex task into smaller doable tasks). **Note: Creating a a software project of even a few hundred lines of code rapidly becomes complex if not overwhelming. The key to success is to take baby steps.**
13. Complete the first chapter of the AP Computer science case study GridWorld.

Assignments: The below assignments are to be completed within the first 10 days of school.

GridWorld: Read Chapter 1; Do exercises 1-4. Note: you will not understand all of the terminology. Don't worry: your not supposed to. However, you'll find that if you don't let the jargon bother you, you can muddle through. In programming, muddling is a highly valued art.

Programming Assignment 4: Write a program which receives three input characters and outputs your printed initials in large size (at least 2 inches high) using the 3 input characters stored in string variables. Note: you need to write the initials on graph paper in order to know how many spaces to put between the string variables.

Example:

Input: "T", "K", "R"

Output:

```
T T T T T   K     K     R R R R
    T       K   K       R     R
    T       K K         R R R
    T       K   K       R     R
    T       K     K     R       R
```

Read Sections: 2.1 to 2.3

Essential Question: What is a computer?

Chapter 1 & 2 (continued): Hardware Basics

(VI Computing in Context)

14. Define the terms bit, byte, kilobyte, megabyte, and gigabyte.
15. State the basic types of gates and give logic tables for each.
    http://scitec.uwichill.edu.bb/cmp/online/P10F/logic_gates2.htm

    - **and**
    - **or**
    - **xor**

15. Correctly identify and describe the functions of the following components.
    - **CPU**
    - **RAM**--primary memory, usually volatile
    - **Hard drive**--secondary memory, non volatile
    - **Power Supply and its highest output voltage (12 v)**--static charge must be above 10,000 volts (more than enough to "zap" electronics) to be felt
    - **Mother board/bus**--how the CPU communicates with the computer's parts
    - **Video Card/monitor and keyboard**--how the computer communicates with you
    - **USB, Parallel, and Serial ports**--how the computer communicates with peripherals, such as printers
    - **Network card**--how the computer communicates with other computers

16. Describe the function of ROM and the BIOS.
17. Describe the function of a UPS.

18. Name the 2 most common I/O devices.
19. Compare single user computer systems with Networks.

- **stand-alone applications**--word processors, spread sheets, etc.
- **information sharing and communication**--various mark up languages such as html,
- **distributed software**--applets, java scripts, server side software, etc
- **cluster computing**--Beowulf systems, SETI

Read Sections: 1.1 to 1.4

Test: Chap 1 & 2 Objectives 1-19

Essential Question: How are numbers, letters, and colors represented in computers?

Chapter 1 & 2 (continued): How Information is Represented

(III Program Analysis)

1. Describe the following number systems: binary, decimal, and hexadecimal
2. Convert binary to decimal numbers.
3. Convert binary to hexadecimal numbers.
4. Convert hexadecimal to binary.
5. Convert hexadecimal to decimal numbers.
6. State how characters are represented inside computers and give the name of the 2 types of code.

Code Type	Bytes	Possible Characters
ACII	1	256
Unicode (used in Java)	2	65,536

8. Describe how colors are commonly represented
9. Given a quantity of bits specify the largest number which can be represented.
10. Given a number of bits specify the largest number of objects or colors which can be specified.

Read Sections: 1.5 to 1.7
Homefun: Complete the number conversion and Unicode worksheet

Essential Question: What is the most important part of a piece of software's life cycle?

Chapter 1 & 2 (continued): Software Design Basics

(I Object-Oriented Program Design)

11. Use the two major computer program development tools.

    - Pseudo code
    - Flow Charts

12. Contrast procedural and object oriented programming.
13. List and explain the 6 phases of the Software System Life Cycle.
    - Analysis
    - Design
    - Coding
    - Testing/verification
    - Maintenance
    - Obsolescence
14. Describe top-down design.
15. Describe key strategies in the test phase.

    - **isolation testing**--classes and libraries are tested separately
    - **boundary condition testing**--boundaries are identified and used for generating test data.
    - **integration testing**--all of the softwares components are combined and tested as a unit

16. State the type of program used for delivering Java programs over the internet.
17. Write a simple applet.

Homefun: Read Chap. 2, Exercises 1, 6, p. 38
Programming Assignment: Exercise 14, 16

Test: Chap 1 & 2 Objectives 1-17

Essential Question: If a program has several million lines of code and is being worked on by people around the globe 24 hours a day, how do you prevent one part of the code from interfering with the other parts?

Chapter 3: Introduction to Classes & Objects

(I Object-Oriented Program Design, II Program Implementation)

1. Correctly use the following terms and be able to identify and use them in programming:

    - API (Applications Programming Interface)--**the ultimate source of information for reusable Java code.**
    - packages
    - fully-qualified name - **Library.package.class example: javax.swing.JButton (means JButton class)**
    - wild card - *
    - methods (instance, static)
    - parameters
    - arguments
    - constructor (called with new operator when the object is created and allocated RAM)

    - field (primitives or objects)
    - instance
    - object
    - static
    - event
    - listener
    - interface
    - classes

2. Explain how inheritance works and why it is used.
    - **superclass**
    - **subclass**--"is-a" relationship

3. Explain the "has-a" relationship.
4. Explain the term "primitive data types" and give examples.
5. Name four classes contained in the java.lang package which is automatically imported into your java programs.

    - **System**
    - **Math**
    - **Object**
    - **String**

Homefun: Read Chap. 3
Write brief definitions for the above terms in objectives 1 and 2.
Programming Assignment Exercise 9, 10, 12

Programming Assignment: Write a program which uses command line input to input a single dimension in centimeters. Use this dimension to calculate and output the surface area and volume of a cube, sphere, and cylinder along with the correct units. The radius

and height of the cylinder are equal to the dimension. Use a separate method for each calculation.

Essential Question: How does an algorithm compare to a mathematical model in physics or engineering?

Chapter 4: Algorithms
(III Program Analysis)

6. Define the term algorithm.
7. Correctly use recursion
   Programming Assignment: Write a program that inputs an integer n using command line input and uses it to calculate n factorial using recursion.

8. Determine the number of iterations a given recursion will have.
9. Define list in abstract terms.
10. Define traversal.

Homefun: Read Chap. 4; Exercises 1, 7, 9,
Programming Assignment: Exercise 10
Test: Chap 3 & 4 Objectives 1-10

Essential Question: What's the difference between style and syntax??

Chap. 5: Java Syntax and Style

(II Program Implementation, III Program Analysis)

1. Correctly use the three forms of comments:

   - // Single line comment
   - /* One or more lines of comments*/
   - /** Javadoc comments*/

2. Identify reserved words (p. 107).
3. Correctly use the naming conventions for classes, methods, and fields.

   - Capitalize the first letter of classes but not methods or fields.
   - **The first character must be a letter & have no spaces in it.**
   - Names can include letters, numbers or the underscore_.
   - Names should be descriptive.
   - Method names = verbs, field = nouns
   - Constants use all capitals.

4. Correctly indent programs.

Homefun: Read Chap. 5; Exercises 3, 4, 5, 7, 8, 10
Programming Assignment: Lab 5.6

Essential Question: Why did you learn to find remainders in grade school instead of going straight to long division ?

Chapter 6: Data Types, Variables, and Arithmetic

(II Program Implementation, III Program Analysis)

5. Understand the meaning and use of the equal sign in Java.

    - Can have only one field or variable on the left side
    - Means "replaced by", not equals

6. Correctly declare fields and local variables.
7. Correctly initialize fields and local variable.
8. State the default value used for initializing fields.
9. State the default value used for initializing local variables.

12. State the eight types of primitive data types and their sizes in bytes.
13. Explain why the largest size of and integer of long can be a major issue.
14. Explain the limitations of precision on floating point data types and why these limitations can produce round-off errors.
15. State the primitive data types which do not have a true decimal point and explain why this can be a problem.
16. Correctly Use:

    - literal constant - **letters in single quotes and numbers**
    - symbolic constant - **declared and initialized using final**
    - escape sequences (p.146, \n newline, \t tab, \\slash, etc.)

17. Understand the term scope (p. 133). **Note: the concept of scope is incredibly important to programming in Java. You must consider it whenever you create a variable. As a matter of style, variables should be declared at the top limit of their scope.**

    - inside a class
    - inside {}
    - inside a method

18. Correctly use compound assignment and incremental operators.
19. Correctly convert numbers and objects into strings.

Homefun: read Sections 6.1 to 6.5; Exercises 1- 7 p.146-147

20. Perform integer division.
21. Truncate and round numbers using integer division.
22. Cast variables.

Example:
```
int a, b;
double c;
c = (double) a / (double) b;
```

23. Use literal constants as either int or doubles (**Example:** 2, 2.0).
24. Correctly use various arithmetic operators including:

- modulus (%)
- compound assignment operators (/=, +=, -=, *=, %=)
- increment/decrement (y++, ++y, y--, --y)

25. Correctly use the order of operation for arithmetic.

1. parentheses
2. division, multiplication, modulus
3. addition, subtraction

Homefun: Read Sections 6.6 to 6.10; Exercises 8, 10, 11 p.147-148
Programming Assignment: Lab 6.8, 6.9, Exercises 12, 13 p.167

Essential Question: How does a progressive income tax system work and is it a good idea?

Tax code program:

Input: Using the command line input income in dollars. (This needs to be changed to cents inside the program.)

Output:

- **Taxable income in dollars**
- **Tax due in dollars** (round tenths of cents upward)
- **Nominal tax rate in % of income**

NTR = (taxDue) / (income) *100

Description: The program will round any tenths of a cent upward. "If" statements are not allowed. The program will use only algorithms to calculate taxes. It will use the following progressive tax table:

Income ($)	Tax Rate	Comments
0 to 19,999.99	00 %	This part of income is never taxed
20,000 to 29,999,99	25 %	Only income above $19,999.99 is taxed at 25%
30,000 +	35 %	Only Income above $29,999.99 is taxed at 35%. Note this is an additional 10% above the 25% tax that kicks in at $29,999.99.

Test: Chap 5 & 6 Objectives 1-25

Mr. Rogers AP Computer Science A - Second Quarter Objectives

Essential Question: Do we live in a binary world?

Chapter 7: Boolean Expressions and Conditional Control

(II Program Implementation, IV Standard Data Structures)

1. Define conditional control.
2. Correctly use both if and if-else statements. (Be as one with the four Common if-else Errors on p. 156.)
3. Draw flowcharts for both if and if-else statements.
4. Correctly use boolean relational operators (p. 179, 181). == , > , < , >= , <= , !=
5. Correctly use boolean logical operators (p. 179, 181). && , || , !
6. Given a set of logic gates write the associated boolean expression (p. 3).
7. Write truth tables for "and", "or", and "xor" gates.
8. Evaluate complex boolean statements using correct order of operators (p. 184).
9. Correctly describe and use short-circuit evaluation (p. 184).
10. Correctly use if-else-if tables for menus and selection processes like tax tables. (p. 187)
11. Correctly use switch statements. (Be as one with the six points on p. 197)

Homefun: Exercises 1-7
Programming Assignments: Lab 7.10, Lab 7.12; exercises 14, 17

Tax Code Program II: rewrite the tax program entirely with if-else-if tables

Test: Chap 1 & 2 Objectives 1-29

Essential Question: How can we access and perform similar manipulations on millions of volatile pieces of similar information without having to create millions of lines of code?

Chapter 8 Iterative Statements (Loops)

(II Program Implementation)

1. State the 3 elements which must be present for any loop (p. 299).

    - **initialization**
    - **testing**
    - **incrementing**

2. Correctly use "while" loops.
3. Correctly use "for" loops.
4. Correctly use "do-while" loops.
5. State which loop always runs at least once.
6. State the most appropriate loop(s) to use when the number of iterations are known and when they are not known.
7. Appropriately use break and return statements for ending loops.
8. Correctly use loops for manipulating values in arrays.
9. Correctly use nested loops.
10. State the number of times each loop runs when using loops or nested loops.
11. Be aware that a break statement in an inner loop will only break out of the inner loop.
12. Use loops for the following programs:

    - even/odd numbers
    - factorials
    - fibonacci numbers

13. Be able to read and modify the Marine Biology Case (MBCS) study

Homefun: Exercises 8, 9, 13
Programming Assignment: Lab 8.6; Exercise 10, 11, 12

Test: Chap 8 Objectives 1-13

Essential Question: Why use objects instead of just using variables?

Chapter 9: Implementing Classes and Using Objects

(II Program Implementation, IV Standard Data Structures)

1. Properly declare a class.
2. Explain why fields are usually declared private and methods public.
3. Correctly declare and initialize (generally with the new operator) objects
4. Be as one with the nature of methods.

    - must be called to run
    - run sequentially from top to bottom unless branched
    - can be called repeatedly
    - when finished returns control of the program to the place the method was called.
    - contain a small highly focused program or algorithm.
    - modular and reusable
    - defined in a class
    - a form of **control abstraction** in which a single line of code, calling the method, can be used to perform a useful function without having to know all the details of how the method's code is written.
    - static (called by a class name) or instance (called by an instance)

5. Correctly write the header which declares a method inside a class.

    access (public or private) returnType methodName (type parameterName1, ... type parameterNameN)

7. Explain the terms precondition and post condition as they relate to methods and procedural abstraction.
8. Explain how assertions and runtime exceptions are used in debugging, especially for reusable code.
9. Correctly use comments with method declarations so that a programmer could use the method without knowing the details about how the code is implemented.
10. Explain the nature of parameters.

    - creates a local variable inside the method
    - used for passing values into the method
    - when using objects can also pass a value out of the method

9. Explain the difference between an argument and a parameter.
10. Correctly use overloaded methods. (Same name different arguments)

- different numbers or types of arguments
- return type must be the same.

11. Correctly use constructers.

    - name is identical to the class name
    - no return type, not even void
    - can be overloaded
    - not required

12. Correctly use copy constructors.
13. State when the default constructer runs and how it initializes fields.

    - numbers = 0
    - booleans = false
    - references = null

14. Correctly use the "final" reserved word.
15. Correctly initialize objects and use the new operator (p. 224).

Homefun: read Sections 9.1 - 9.5; Exercises 1-6

16. State which type of fields can be accessed and what type methods can be called using a static method. (p. 223).

    - can access static field - **only one value & one memory address**
    - can call static method
    - can access instance field of a **different class**
    - can call instance method of a **different class**

17. State which type of fields can be accessed and what type methods can be called using an instance method (p. 223).

    - instance field - **Many values & memory addresses.** A field has one value per object
    - static field - **Only one value & memory address**
    - instance method
    - static method

18. Correctly call both static and instance methods (p. 226).

    - Use class name & dot operator when calling **static** methods from another class.
    - Use object & dot operator when calling **instance** methods from another class.
    - Class name & dot operator are optional when calling **static** methods in a class.

- Use "this" & dot operator or nothing when calling **instance** methods in a class.

19. Compare the two ways to pass arguments to methods and constructers (p. 230).

    - Primitive data types are always passed as values
    - Objects are always passed as references.

20. Correctly use return statements.

Homefun: Read Sections 8.6 - 8.10; Exercises 8, 9, 13
Programming Assignment: Lab 8.6; Exercise 10, 11, 12

Test: Chap 9 Objectives 1-20

Essential Question: How can we store and manipulate not just numbers but human text in a computer program?

Chap. 10 Strings

(II Program Implementation)

1. Recognize literal strings.
2. Be familiar with the 2 most common of String's 9 constructors.

    - String ()
    - String (String s)

3. Be aware that Strings are immutable objects.
4. Understand how Strings are initialized.

    - String s1 = " "; **//s1 is set to an empty String**
    - String s2 = new String(); **//s2 is set to an empty String**
    - String s3; **//s3 is set to null**

5. Be aware that calling a String method with a String set to null will give run-time errors. (An empty string initialized to " " is not the same thing as a string initialized to null.)
6. Correctly use common String methods (See table on p. 265.)

    - length()
    - charAt (pos) **//Note that characters are counted starting at 0.**
    - substring (pos)
    - substring (fromPos, toPos)

- compareTo (s2) **//returns neg int if s1 <s2, 0 if s1 = s2, pos int if s1>s2**
- compareToIgnoreCase (s2) **//returns neg int if s1 <s2, 0 if s1 = s2, pos int if s1>s2 ignoring case**
- equals(s2) **//returns a boolean true or false**
- equalsIgnoreCase(s2) **//returns a boolean true or false ignoring case**
- indexOf (ch) **//four types**
- lastIndexOf (ch) **//four types**
- trim() **//removes white space characters from beginning and end of string**
- replace (oldChar, newChar)
- toUpperCase()
- toLowerCase()

7. Correctly use the various ways of doing concatenation (adding Strings). (p. 266)

    - yourString += str;
    - yourString = yourString1 + str;
    - yourString.concat (str);

8. Correctly use relational operators with Strings. p. 268)
9. CANNOT use == , != , < , >, <= , >=
10. CAN use equals(s2), equalsIgnoreCase(s2), compareTo(s2) methods, compareToIgnoreCase (s2)
11. Correctly convert numbers into Strings and Strings into numbers.
12. Explain the term wrapper class.
13. Correctly use the Character wrapper class methods. (p.275)
14. Correctly use the StringBuffer class (p. 278).

Homefun: Exercises 1-6;
Programming assignments: Lab 10.8

Test: Chap 10 Objectives 1-13

Essential Question: How does OOP development using Java compare with the process of writing a book using English?

Chapter 11: Class Hierarchies and Interfaces

(I Object-Oriented Program Design, II Program Implementation, III Program Analysis

1. Name 2 reasons why duplicate code is a bad idea. (wasteful, hard to maintain)

2. Explain the purpose of having a hierarchy of classes and how it helps facilitate functional decomposition (breaking down an overwhelmingly complex task into smaller doable tasks).
3. Define polymorphism and explain why it is useful (page 293).
4. Create and use abstract methods (methods that are declared but undefined).

```
Example declaration:
public abstract int tax();  // Note that here is no code, not even
brackets.
```

5. Create abstract classes (classes with one or more abstract methods).

```
Example declaration:
public abstract class People {
    . . .
}
```

6. Correctly declare an interface.

```
Example declaration:
public interface Career {
    . . .
}
```

7. Define the term concrete class.
8. Be aware that all classes automatically extend the class Object.
9. State the difference between an interface, abstract class, and concrete class.
10. Be aware that Java does not allow the creation of objects (instances) of an abstract class or an interface.
11. Describe the reasons for using an interface.
12. Be aware that if a class implement an interface (and it can implement as many as it wants), it must supply all the methods specified in the interface.
13. Write programs that extend a given class using inheritance.
14. Write programs that implement interfaces.
15. Explain how polymorphism applies to interfaces.

Homefun: Exercises 1-5

Design Assignment: Design a program using Bluej called Mr. Rogers High School with a Hierarchy of classes and at least one interface. Specify major fields and methods. Write headers and descriptors for all classes and method. Other than headers, your design should include no code, but should include the Bluej box diagram.

Test: Chap 11 Objectives 1-15

Essential Question: How can we store millions of volatile pieces of similar information without having to create millions of variables in a computer program?

Chap. 12 Arrays and ArrayLists

(IV Standard Data Structures, V Standard Algorithms)

1. Correctly declare and initialize arrays.

    - int scores [ ] = new int[5];
    - int scores [ ] = { 93, 67, 99, 87, 91};

2. State the default values when initializing arrays

    - numbers = 0
    - boolean = false
    - objects = null

3. Know that once declared and initialized, the size of an array cannot be changed.
4. Know that the elements of arrays are numbered starting with 0. These elements can be randomly accessed using the indices or subscripts.
5. Correctly use the array length field.
6. Be aware that arrays are always passed to methods by reference.
7. Correctly declare and initialize 2D arrays.

```
final int ROWS = 2;
final int COLS = 3;
double name2DArray [ ] [ ] = new double [ROWS][COLS];
   or
double name2DArray [ ] [ ] =
{
      {0.0, 2.7, 5.6},
      {4.3, 1.4, 7.2}
};
```

8. Be aware that the first dimension in a two dimension array is considered the number of rows while the second dimension is the number of columns.
9. Correctly use the length field to find the number of rows and columns.

    - twoDArray.length gives the number of rows
    - twoDArray[0].length gives the number of columns

10. Be aware that arrays with more than 2 dimensions can be declared.
11. Correctly use parallel arrays.
12. State the advantage of using ArrayList.
13. Be familiar with ArrayList's constructors and methods (see page 331).

14. Be aware of the pitfalls of using ArrayLists.
15. Describe how classes and arrays are key elements used to form data structures.
16. Describe why abstraction is an important theme in data structures.
17. Be as one with Chapter 2 of GridWorld.

Homefun: Exercises 1-5;
Programming assignments: Lab 12.9, 12.11

12.1 Simple Data Base Program

Create a program with a class called People. In the main method create 3 parallel arrays: 2 string arrays holding 3 first names and 3 last names plus an IQ array with the corresponding IQs. Use an output method to alphabetize by last name, concatenate the first and last names, and output the data in two columns as shown in the example:

Name	IQ
Bob Arnold	90
Juan Smith	200
Betty Taylor	142

12.2 2-D Array Output

Create a program with a class called Alphabet with 3 private fields of 2-dimentional arrays: each will contain a large-sized letter respectively A, B, and C. Use a separate method to output the letters alphabetically in a column with A at the top.

Note: You need to write the letters on graph paper or in an Excel spread sheet in order to understand how many spaces are needed. To understand how the arrays should be initialized look at chapter 10 objective 7 above. You do not need to reproduce the grid lines in your program's output.

	0	1	2	3	4	5	6	7	8	9	10
0						A					
1					A		A				
2				A				A			
3			A	A	A	A	A	A	A		
4		A								A	
5	A										A

GridWorld: Read Chapter 2, Exercises 1-4

Test: Chap 12 Objectives 1-16

Essential Question: How can we search and sort millions of volatile pieces of similar information without having to create millions of lines of code?

Chapter 13: Searching Sorting and Other Array Elements

(V Standard Algorithms)

1. **Sequential and binary searches:** Write the code .
2. Given a set of data records be able to pick the above search method which will be likely to run the fastest.
3. Write the code which can find the largest or smallest element in an array.
4. Correctly use loops for the following:
    - **traversals**
    - **insertions**
    - **deletions**
5. **Selection sort:** Write the code .

A. Initialize an int variable n to the largest index in the array.
B. Find the biggest in the first (n) elements array.
C. Swap the biggest for the element n.
D. Perform n - -
E. Repeat while n >= 1.

5. Be as one with the descriptions of the three sort methods mentioned on page 342

    - **selection sort**
    - **insertion sort**
    - **merge sort**

6. Given a set of data records be able to pick the above sort method which will be likely to run the fastest.
7. Name the search and sort algorithms which could be categorized as divide and conquer techniques.
8. Be aware of the methods associated with the Array class.
9. Be as one with chapter 3 of Gridworld.

Homefun: Exercises 1-5,
Programming assignments: Lab 12.5 Rotate an Array, Lab 12.9 Benchmarks, 12a,b,c (p. 319).
12.1 Array Search

1. Create a program with an integer array containing 20 random numbers. Use code to initialized the array with random integers from 0 to 100. Using command line input search the array for a specific number. Output the entire array, the highest value, the lowest value, and the index number where the number you searched for was found or a statement that the number was not found.
2. Add a selection sort (write the code, see p. 332) to the above and output the sorted array by creating a Sort class with selection sort method in it.
3. Add the code for a binary search which will then search for the element specified earlier in command line input.

Test: Chap 13 Objectives 1-6

Essential Question: How does Java allow a program to read and write to the hard drive?

Chapter 14: Streams and Files

Note: Chapters 14 covers items not typically not found on the AP exam but will be useful in creating your personal project.

1. State the 2 major categories of files.

    - binary
    - ASCII or Unicode (text)

2. Describe how the lines in text files are ended (CR+LF).
3. Be as one with the following vocabulary: **buffer, stream, random-access file.**
4. Be aware that a stream can be opened for input or output but not both at the same time.
5. Be aware that a random access file can start reading or writing at any point in the file. It can be opened for both reading and writing at the same.
6. Be aware that text files tend to have different lengths and are usually treated as streams while binary files of fixed length are treated as random-access files.

Note: Chapters 15, 16, and 17 deal with graphics, GUI components, mouse, keyboard, sounds and images. These items are typically not found on the AP exam but will be useful in creating your personal project.

Assignment: Continue working on your personal project.

GridWorld Part 3: GridWorld Classes and Interfaces
In Class Group Activity: We will do the group activity at the end of Chapter 4 in class

HomefunRead Chapter 3 , Do you know? Set 6

Mr. Rogers AP Computer Science A- Fourth Quarter Objectives

GridWorld Chapter 4:
Group Activity: We will do the group activity at the end of Chapter 4 in class
Read Chapter 4 Interacting Objects, Exercises 1-6,

Assignment: Complete your personal project

AP Exam Free Response Preparation

Homefun (turn in 3 per week):

1. Answer all the questions in the free response portion of the 2004 and the 2005 AP Computer Science A exam except for the Marine Biology Case Study question.
2. Grade and score the answers to your partner's answers to the above according to the AP answer guide. Sign the problems you grade.
3. Be familiar with the java library classes in the A-level AP Java Subset (see the College Board AP Computer Science A Site)

Problem sets can be found online at the AP Computer Science A section of the American College Board site or will be provided by Mr. Rogers.

AP Computer Science A Exam % Right on Exam	
AP Score	**2004**
5	**79**
4	**61**
3	**49**

AP Exam Multiple Choice Preparation

In-Class Tests: A multiple choice AP type test will be given approximately once per week starting near the end of March (3 tests total). These will count 100 points each toward your grade. They will be curved to approximate an AP grading system. At worst 50% correct will be a "C". In addition, each student will receive an estimate from 1 to 5 of their future grade based on each test.

Take-Home Tests: A minimum of 2 multiple choice AP-type take-home tests will be given. These will count 50 points each and be curved but not as generously as the in-class tests. Your work is to be turned in on each question. You may collaborate with other students and may compare answers but only if each person has actually worked the problem and written down their work. Allowing a student to simply copy your answers is strictly forbidden and may result in a grade of zero for both of the students involved.

The Good News: The highest in-class test will be cloned to help compensate for having a bad day. Take-home tests will not be cloned.

AP Exam Self Study

As mentioned on the first page, you will not reach your potential on the AP test without a lot of self study. This AP test study should start in December. At that time you should start practicing the multiple choice practice tests in the Barron's AP review book. These questions are close to those actually found on AP tests.

Essential Question: Do computing professionals make life and death decisions?

The Social and Ethical Considerations in Computing

(VI Computing in Context)

1. System reliability--the horror stories: radiation burned cancer patients, phone system crashes, etc.
2. The right to privacy--you would not walk into your neighbor's house merely because the door was open.
3. Legal issues

    - software piracy--**When you pirate software you are stealing from every single user who has paid for the product. This eventually shows up as an added cost in a wide variety of products other than just software related ones.**
    - intellectual property
        - copyright law--**anything you write including code is automatically copyrighted as soon as it is written. Copyright law is very strong protection.**
        - trademarks
        - patents
    - identity theft
    - e-mail scams--**you've won a million dollars, yeah right.**
    - vandalism

- viruses--**the price you pay for antivirus software is only part of the continuing cost you are forced to pay because of individuals who write viruses.**

4. Social and ethical ramifications

    - spyware
    - SPAM
    - chain letters--**when you politely complain to the sender use "reply". Never use "reply to all" it compounds the problem**.
    - search engine issues--**when information from one site is copied and posted on another, without giving a reference to the original source, it becomes impossible to find the original source and evaluate the reliability of the information.**

5. Define responsible use of computers.

Assignment: Pick a topic from any of the above items and prepare a 5 minute PowerPoint presentation to be delivered in class after the AP exam.