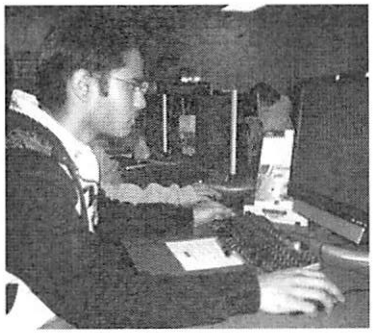
Approved by California Department of Education
Accredited by Western Association of Schools and Colleges

UC certified "a-g"



Berkeley Technology Academy (BTA)
Computer Programming/Computer Science

Curriculum

BTA Computer Programming/Computer Science

(AP Computer Science A – Java)

PROGRAM GOALS

The goal of the program is to develop students' academic and technical skills, preparing them for college, advanced training, employment, and opportunities for promotion.

COURSE TITLE	BTA Computer Programming/Computer Science
CBEDS TITLE/NO.	Computer Operations/Computer Science – 4601
JOB TITLE/ONET CODE:	15-1021.00 - Computer Programmer 15-1031.00 - Computer Software Engineer
INDUSTRY SECTOR:	Information Technology Industry Sector
PATHWAY:	Programming and Systems Development Pathway

I. COURSE DESCRIPTION

In this class, students learn Java, an object-oriented programming language. Instruction includes problem solving and algorithm development, data structures and design, and the use of logic and formal methods. Students may prepare for the AP Computer Science A exam (optional). In addition to instructional time, students need to spend at least three hours per week practicing their programming skills.

This course is designed to provide the skills for an entry-level position in computer programming or provide a foundation for further studies in computer science at the college level. Integrated throughout the course are career technical education standards, which include basic academic skills, communication, career planning, technology, problem solving, safety, responsibility, ethics, teamwork, and technical knowledge.

Prerequisites:

- Algebra I
- Basic computer skills
- Student must be at least 16 years old or a Junior or Senior

Hours: Students may receive up to **180** hours of classroom instruction

Date revised: 10/26/09

UC "a-g": This course is certified by the University of California as an "a-g" course in the "**g-Elective**" category for Contra Costa County ROP. *High schools must include this course on their own "a-g" list in their annual on-line update through the UC Web site.*

II. STUDENT PERFORMANCE OBJECTIVES

A. Course Objectives—Students will:

1. Learn basic Java programming language *(ESLR #1, #2)*

2. Design and implement computer-based solutions to problems in a variety of application areas by writing, running, and debugging computer programs *(ESLR #3)*

3. Build on previously acquired mathematical reasoning skills to develop problem-solving programming methodology *(ESLR #3)*

4. Use and implement commonly-used algorithms and data structures *(ESLR #1, #2)*

5. Develop and select appropriate algorithms and data structures to solve problems *(ESLR #1, #2)*

6. Code fluently in an object-oriented programming language (students are expected to be familiar with the standard Java library classes from the AP Java subset) *(ESLR #1, #2, #3)*

7. Read and understand a large program consisting of several classes and interacting objects; students should be able to read and understand a description of the design and development process leading to such a program *(ESLR #1,#2, #3)*

8. Identify the major hardware and software components of a computer system, their relationship to one another, and the roles of these components within the system *(ESLR #3)*

9. Enhance written communication skills by recognizing the importance of programming documentation *(ESLR #1)*

10. Recognize the ethical and social implications of computer use *(ESLR #4)*

B. Expected School-wide Learning Results (ESLRs) for ROP:

1. Demonstrate effective skills in oral and written *communication.*

   - Speak clearly using professional and industry-specific terminology
   - Develop appropriate listening, speaking, and presentation skills
   - Use technology to enhance communication
   - Read and comprehend industry-related material
   - Write effectively in a variety of different formats

2. Demonstrate *job skills* and the behavior and work ethic valued by employers.

   - Use technology to enhance work performance
   - Acquire industry-specific competencies
   - Meet occupational safety standards
   - Demonstrate appropriate business ethics and etiquette
   - Identify short-term and long-range career goals
   - Demonstrate organizational skills such as goal setting and time management

3. Demonstrate the ability to be critical, complex, and creative *thinkers.*

   - Brainstorm and discuss ideas with others
   - Access resources; organize and analyze information

- Process and apply knowledge to new situations
- Demonstrate problem-solving, computational, and research skills

4. Work productively both as individuals and as *team members.*

   - Demonstrate initiative and resourcefulness
   - Brainstorm and collaborate with others
   - Demonstrate the ability to assume a leadership role
   - Give and receive constructive feedback

III. COURSE OUTLINE

Course Outline	Career Technical Education Standards	Suggested Activities/Assessment
1. **Orientation** *(10 hrs)* A. Classroom procedures and safety unit 1. Review of safety and guidelines 2. Ergonomics and computer use B. Overview of the field of computer science and computer programming	**Standards for the Programming and Systems Development Pathway of the Information Technology Industry Sector** **Foundation Standards** **6.0 Health and Safety** Students understand health and safety policies, procedures, regulations, and practices, including the use of equipment and handling of hazardous materials: 6.3 Understand the environmental and ergonomic risks associated with the use of business equipment and the financial impact of an unsafe work environment. **4.0 Technology** Students know how to use contemporary and emerging technological resources in diverse and changing personal, community, and workplace environments: 4.1 Understand past, present, and future technological advances as they relate to a chosen pathway. 4.2 Understand the use of technological resources to gain access to, manipulate, and produce information, products, and services. 4.5 Know procedures for maintaining secure information, preventing loss, and reducing risk.	o Orientation lecture o Lecture and demonstration of safety material o Lecture on the history and evolution of computer programming languages

2. Object-oriented program design *(30 hrs)* A. Program design 1. Read and understand a problem description, purpose, and goals 2. Apply data abstraction and encapsulation 3. Read and understand class specifications and relationships among the classes a. "is-a" relationship b. "has-a" relationship 4. Understand and implement a given class hierarchy 5. Identify reusable components from existing code using classes and class libraries B. Class design 1. Design and implement a class 2. Choose appropriate data representation and algorithms 3. Apply functional decomposition 4. Extend a given class using inheritance C. Advanced topics — optional 1. Specify the purpose and goals for a problem 2. Decompose a problem into classes a. Define relationships b. Define responsibilities 3. Design and implement a set of interacting classes 4. Design an interface 5. Choose appropriate advanced data structures and algorithms	**Pathway Standards** **D3.0 Students understand the creation and design of a software program:** D3.1 Analyze customers' needs and requirements for software. D3.2 Know how specifications and codes are developed for new and existing software applications. D3.3 Understand the abstract organization of information and how programs maintain the properties of the data structure while they perform such operations as search, insert, or load-balancing. D3.4 Know multiple ways in which to store, retrieve, and access information. D3.5 Understand how to track software versions. **Academic Standards** **2.0 Communications** Students understand the principles of effective oral, written, and multimedia communication in a variety of formats and contexts. **2.1 Reading** 2.4 Synthesize the content from several sources or works by a single author dealing with a single issue; paraphrase the ideas and connect them to other sources and related topics to demonstrate comprehension. 2.6 Demonstrate use of sophisticated learning tools by following technical directions (e.g., those found with graphic calculators and specialized software programs and in access guides to World Wide Web sites on the Internet).	o Lecture, demonstration, and class discussion: The overall goal for designing a piece of software (a computer program) is to correctly solve the given problem. At the same time, this goal should encompass specifying and designing a program that is understandable, can be adapted to changing circumstances, and has the potential to be reused in whole or in part. The design process needs to be based on a thorough understanding of the problem to be solved. o Develop the parts of a program when given its design. This includes an understanding of how to apply data abstractions (classes and arrays). o Describe the inheritance and composition relationships among the different classes that comprise a program. o Implement a class inheritance hierarchy when given the specifications for the classes involved—which classes are subclasses of other classes. o Given a design for a class, either their own or one provided, students should then be able to implement the class; extend a given class using inheritance, thereby creating a subclass with modified or additional functionality. o Write a class that implements an interface. o Design a program to develop a solution that includes the following: • Appropriate use of inheritance from another class using the keyword "extends" • Appropriate implementation of an interface using the keyword "implements" • Declaration of constructors and methods with meaningful names, appropriate

	Academic Standards **1.1 Mathematics—Algebra I** 1.1 Students use properties of numbers to demonstrate whether assertions are true or false. 5.0 Students solve multistep problems, including word problems, involving linear equations and linear inequalities in one variable and provide justification for each step. 13.0 Students add, subtract, multiply, and divide rational expressions and functions. Students solve both computationally and conceptually challenging problems by using these techniques. 15.0 Students apply algebraic techniques to solve rate problems, work problems, and percent mixture problems. 24.1 Students explain the difference between inductive and deductive reasoning and identify and provide examples of each. 24.2 Students identify the hypothesis and conclusion in logical deduction. 24.3 Students use counterexamples to show that an assertion is false and recognize that a single counterexample is sufficient to refute an assertion. 25.1 Students use properties of numbers to construct simple, valid arguments (direct and indirect) for, or formulate counterexamples to, claimed assertions. 25.2 Students judge the validity of an argument according to whether the properties of the real number system and the order of operations have been applied correctly at each step.	parameters, and appropriate return types • Appropriate data representation • Appropriate designation of methods as "public" or "private" • All data declared "private" • All client accessible operations specified as "public" methods ○ Programming assignment—*Using a Date Class* (download from "Nifty Assignments" at AP Central)

Course Outline	Career Technical Education Standards	Suggested Activities/Assessment
3. **Program implementation** *(30 hrs)* A. Implementation techniques 1. Methodology a. Object-oriented development b. Top-down development c. Encapsulation and information hiding d. Procedural abstraction B. Programming constructs 1. Primitive types vs. objects 2. Declaration a. Constant declarations b. Variable declarations c. Class declarations d. Interface declarations e. Method declarations f. Parameter declarations 3. Console output a. System.out.print/println 4. Control a. Methods b. Sequential c. Conditional d. Iteration e. Understand and evaluate recursive methods C. Java library classes • See attached Appendix A and B	**Pathway Standards** **D1.0 Students understand the strategies necessary to define and analyze systems and software requirements:** D1.1 Develop information technology-based strategies and project plans to solve specific problems. D1.3 Know the effective use of tools for software development. D1.4 Know the software development process. **D2.0 Students understand programming languages:** D2.1 Know the fundamentals of programming languages and concepts. D2.2 Compare programs by using control structures, procedures, functions, parameters, variables, error recovery, and recursion. D2.3 Understand digital logic, machine-level representation of data, memory-system organization, and use of assembly-level programming architecture. **Academic Standards** **2.2 Writing** 2.6 Write technical documents: a. Report information and convey ideas logically and correctly. b. Offer detailed and accurate specifications. c. Include scenarios, definitions, and examples to aid comprehension (e.g., troubleshooting guide).	o Lecture, demonstration, class discussion: Classes that fill common needs should be built so they can be reused easily in other programs. Object-oriented design is an important part of program implementation. o Explain several ways to handle input and output in an object-oriented approach to programming, including console-based character I/O, graphical user interfaces, and applets. o Demonstrate an understanding of the concept of recursion and to trace recursive method calls o Recognize the possibilities of reusing components of one's own code or other examples of code in different programs. o See attached Appendix A "AP Computer Science Java Subset" o See attached Appendix B "Standard Java Library Methods Required for AP Computer Science A" o Activity: *Recursion Worksheets* (see Appendix B of *Teacher's Guide)* o Activity: *AP Computer Science Project I* (see Appendix B of *Teacher's Guide)* o Programming Assignment: *Hailstone Sequence;* download from "Nifty Assignments" at AP Central o Programming Assignment: *Cat and Mouse;* download from "Nifty Assignments" at AP Central

Course Outline	Career Technical Education Standards	Suggested Activities/Assessment
4. **Program analysis** *(30 hrs)* A. Testing 1. Test classes and libraries in isolation 2. Identify boundary cases and generate appropriate test data 3. Perform integration testing B. Debugging 1. Categorize errors: a. Compile-time b. Run-time c. Logic 2. Identify and correct errors 3. Employ techniques such as: a. Using a debugger b. Adding extra output statements c. Hand-tracing code C. Understand and modify existing code D. Extend existing code using inheritance E. Understand error handling 1. Understand runtime exceptions F. Reason about programs 1. Pre- and post-conditions 2. Assertions G. Analysis of algorithms 1. Informal comparisons of running times 2. Exact calculation of statement execution counts H. Numerical representations and limits 1. Representations of numbers in different bases 2. Limitations of finite representations a. Integer bounds	**Pathway Standards** **D4.0 Students understand the process of testing, debugging, and maintaining programs to meet specifications:** D4.1 Know the steps involved in the software-testing process. D4.2 Know the methodologies of program maintenance to preserve intended program applications and the operation of scheduled batch jobs and real-time jobs. **Academic Standards** **1.2 Science—Investigation and Experimentation** 1.a Select and use appropriate tools and technology (such as computer-linked probes, spreadsheets, and graphing calculators) to perform tests, collect data, analyze relationships, and display data. 1.d Formulate explanations by using logic and evidence. **2.2 Writing** 2.3 Write expository compositions, including analytical essays and research reports: a. Marshal evidence in support of a thesis and related claims, including information on all relevant perspectives. b. Convey information and ideas from primary and secondary sources accurately and coherently. c. Make distinctions between the relative value and significance of specific data, facts, and ideas. d. Include visual aids by employing appropriate technology to organize and record information on charts, maps, and graphs.	o Lecture, demonstration, and class discussion: The analysis of programs includes examining and testing programs to determine whether they correctly meet their specifications. It also includes the analysis of programs or algorithms in order to understand their time and space requirements when applied to different data sets. o Debug a program using these techniques: hand-tracing code, adding extra output statements to trace the execution of a program, or using a debugger to provide information about the program as it runs and when it crashes. Experiment with available debugging facilities. o Make informal comparisons of running times of different pieces of code by counting the number of loop iterations needed for a computation. o Write a report on a case study o Case Study of Marine Biology Simulation, (download from "Nifty Assignments" at AP Central College Board Web site.) o Case study of GridWorld (download files from the AP Central College Board Web site) o See current AP Computer Science Case Study (download from AP Central) for activities such as the following: • modifying the procedural and data organization of the case study program to correspond to changes in the program specification • extending the case study program by writing new code (including new methods for existing classes, new subclasses extending existing classes, and new classes) • evaluating alternatives in the

b. Imprecision of floating-point representations c. Round-off error I. Advanced topics – optional 1. Throw runtime exceptions 2. Big-Oh notation 3. Worst-case and average-case time and space analysis		representation and design of objects and classes • evaluating alternative incremental development strategies • understanding how the objects/classes of the program interact.
5. Standard data structures *(30 hrs)* A. Simple data types 1. Int 2. Boolean 3. Double B. Classes C. Lists D. Arrays 1. One-dimensional arrays 2. Two-dimensional arrays E. Advanced topics – optional 1. Linked lists (singly, doubly, circular) 2. Stacks 3. Queues 4. Trees 5. Heaps 6. Priority queues 7. Sets 8. Maps	**Foundation Standards** **5.0 Problem Solving and Critical Thinking** Students understand how to create alternative solutions by using critical and creative thinking skills, such as logical reasoning, analytical thinking, and problem-solving techniques: 5.1 Apply appropriate problem-solving strategies and critical thinking skills to work-related issues and tasks. 5.2 Understand the systematic problem-solving models that incorporate input, process, outcome, and feedback components. 5.3 Use critical thinking skills to make informed decisions and solve problems.	○ Lecture, demonstration, and class discussion: Data structures are used to represent information within a program. Abstraction is an important theme in the development and application of data structures. ○ Use the standard representations of integers, real numbers, and Boolean (logical) variables. ○ Become familiar with the Java "String" class and the methods of the "String" class that are listed in the AP Java subset (see Appendixes A and B to *Course Description*, download from AP Central) ○ Activity: *The Mathematical Matrix Class* (See appendix B of *Teacher's Guide*, download at AP Central) ○ Programming Assignment: *Car Rental* (download from "Nifty Assignments" at AP Central) ○ Programming Assignment: *Twelve Days of Christmas* (download from "Nifty Assignments" at AP Central)

Course Outline	Career Technical Education Standards	Suggested Activities/Assessment
6. Standard algorithms *(30 hrs)* A. Operations on data structures previously listed 1. Traversals 2. Insertions 3. Deletions B. Searching 1. Sequential 2. Binary C. Sorting 1. Selection 2. Insertion 3. Mergesort D. Advanced topics - optional 1. Operations on AB-level data structures previously listed (iterators, as well as traversals, insertions, and deletions) 2. Hashing 3. Quicksort 4. Heapsort	**Pathway Standards** **D5.0 Students understand the importance of quality assurance tasks in producing effective and efficient products:** D5.1 Know the standards and requirements for software quality assurance. D5.2 Know common quality assurance tasks and their place in the development process. D5.3 Understand the ways in which specification changes and technological advances can require the modification of programs. D5.4 Know various sorting and searching methods and their comparative advantages. D5.5 Know the characteristics of reliable, effective, and efficient products.	o Lecture, demonstration, and class discussion: Standard algorithms serve as examples of good solutions to standard problems. Many are intertwined with standard data structures. These algorithms provide examples for analysis of program efficiency. o Describe standard algorithms for accessing arrays, including traversing an array and inserting into and deleting from an array. o Explain the two standard searches, sequential search and binary search, and the relative efficiency of each. o Define three standard sorts that are required: the two most common quadratic sorts—Selection sort and Insertion sort—and the efficient Merge sort. o Activity: *Searching and Sorting Worksheets* (see Appendix B of Teacher's Guide) o Activity: *Searching and Sorting Arrays* (see Appendix B of Teacher's Guide) o Programming Assignment: *Guessing Game* (download from "Nifty Assignments" at AP Central) o Practice exams (download from AP Computer Science home page at AP Central) o Algorithm Analysis Research Project (See Appendix B of Teacher's Guide)

Course Outline	Career Technical Education Standards	Suggested Activities/Assessment
7. **Computing in context** *(10 hrs)* A. Responsible use of computer systems 1. System reliability 2. Privacy 3. Legal issues and intellectual property a. Licensing b. Property rights and copyright laws 4. Social and ethical ramifications of computer uses B. Major hardware components 1. Primary and secondary memory 2. Processors 3. Peripherals C. System software 1. Language translators/compilers 2. Virtual machines 3. Operating systems D. Types of systems 1. Single-user systems 2. Networks a. Understanding networking b. Today's world is networked	**Pathway Standards** **D6.0 Students understand the importance of effective interfaces in the interaction between humans and computer systems:** D6.1 Understand how to support access, privacy, and high ethical standards in computing. D6.2 Use knowledge of cognitive, physical, and social interactions to create and design user-friendly computer practices and applications that meet the needs of the market. **Foundation Standards** **8.0 Ethics and Legal Responsibilities** Students understand professional, ethical, and legal behavior consistent with applicable laws, regulations, and organizational norms: 8.2 Understand the concept and application of ethical and legal behavior consistent with workplace standards. 8.3 Understand the role of personal integrity and ethical behavior in the workplace. 8.5 Know how to design systems and applications to allow access to all users. **10.0 Technical Knowledge and Skills** 10.1 Know how to use a variety of business- and industry-standard software and hardware, including major proprietary and open standards.	○ Lecture, demonstration, class discussion: An awareness of the ethical and social implications of computing systems is necessary for the study of computer science. Given the tremendous impact computers and computing have on almost every aspect of society, it is important that intelligent and responsible attitudes about the use of computers be developed as early as possible. The applications of computing that are studied in this course provide opportunities to discuss the impact of computing. Typical issues include: • Impact of applications using databases, particularly over the Internet, on an individual's right to privacy • Economic and legal impact of viruses and other malicious attacks on computer systems • Need for fault-tolerant and highly reliable systems for life-critical applications and the resulting need for software engineering standards • Intellectual property rights of writers, musicians, and computer programmers and fair use of intellectual property ○ Lecture on major hardware components, system software, and types of systems—how it all fits together

Course Outline	Career Technical Education Standards	Suggested Activities/Assessment
8. Career Path *(10 hrs)* A. Exploring careers in computer science and computer programming B. Management of time and resources C. Resumes and cover letters D. Interviewing skills E. Class discussion about desirable on-the-job characteristics, such as responsibility, flexibility, leadership, and teamwork	**Academic Standards** **2.2 Writing** 2.5 Write job applications and résumés: a. Provide clear and purposeful information and address the intended audience appropriately. d. Follow the conventional style for that type of document. **Foundation Standards** **3.0 Career Planning and Management** Students understand how to make effective decisions, use career information, and manage personal career plans: 3.1 Know the personal qualifications, interests, aptitudes, knowledge, and skills necessary to succeed in careers. 3.2 Understand the scope of career opportunities and know the requirements for education, training, and licensure. 3.6 Know important strategies for self-promotion in the hiring process, such as job applications, résumé writing, interviewing skills, and preparation of a portfolio. **7.0 Responsibility and Flexibility** Students know the behaviors associated with the demonstration of responsibility and flexibility in personal, workplace, and community settings: 7.1 Understand the qualities and behaviors that constitute a positive and professional work demeanor. 7.2 Understand the importance of accountability and responsibility in fulfilling personal, community, and workplace roles. 7.3 Understand the need to adapt to varied roles and responsibilities. **9.0 Leadership and Teamwork**	o Discuss the personality traits that might lead to success in the field of computer science o Identify the different careers available in the field of computer science and computer programming o Discuss educational requirements o Invite guest speakers to talk about their educational experience and on-the-job experience o Class discussion on time management, responsibility, flexibility, leadership, and teamwork o Write resumes, cover letters and thank you letters o Practice interviewing skills

Total hours of classroom instruction: 180	Students understand effective leadership styles, key concepts of group dynamics, team and individual decision making, the benefits of workforce diversity, and conflict resolution: 9.1 Understand the characteristics and benefits of teamwork, leadership, and citizenship in the school, community, and workplace settings. 9.3 Understand how to organize and structure work individually and in teams for effective performance and the attainment of goals. 9.5 Understand how to interact with others in ways that demonstrate respect for individual and cultural differences and the attitudes and feelings of others.	

Note: Much of the above material for items #2 - #7 of the "Course Outline" and "Suggested Activities" was obtained from "Computer Science A-Course Description" (May 2010-May 2011), downloaded from the AP Central College Board Web site at http://apcentral.collegeboard.com/apc/public/repository/ap-computer-science-course-description.pdf

References to "Nifty Assignments" can be downloaded at http://apcentral.collegeboard.com/apc/public/courses/teachers_corner/4483.html

IV. METHODS, STRATEGIES AND TECHNIQUES

A variety of strategies and techniques are used to instruct the students, including:

- Direct instruction (lectures, demonstrations, small and large group discussion, selected readings)
- Use of a variety of instructional materials and resources (professional journals, reference materials, textbooks, electronic media)
- Project-based learning
- Collaborative learning opportunities
- Use of community resources including guest speakers and fieldtrips
- Simulations
- Student presentations
- Peer coaching and student mentoring
- Use of technology-based resources
- Hands-on experience
- Group and individual projects

V. ASSESSMENT OF STUDENT PERFORMANCE

Assessment of student performance will include but will not be limited to:

- Tests and quizzes
- Embedded assessments
- Classroom participation, effort, skill mastery and quality of work
- Completion of assignments/portfolio
- Individual projects/group projects
- Punctuality and attendance

VI. BTA/ROP CERTIFICATE REQUIREMENTS

To earn BTA/ROP certification for this course, the student must accomplish the following:

- Complete all of the student performance objectives
- Maintain a **95%** attendance rate
- Demonstrate a positive work attitude

VII. ASSESSED JOB MARKET NEEDS

According to the *Occupational Outlook Handbook, 2008-09 Edition,* published by the U.S. Dept. of Labor, employment of computer programmers is expected to decline slowly, decreasing by 4 percent from 2006 to 2016, due to consolidation and outsourcing. Nevertheless, employers will continue to need some local programmers, especially those who have strong technical skills and who understand an employer's business and its programming requirements.

Although employment is projected to decline, numerous job openings will result from the need to replace programmers who leave the labor force or transfer to other occupations. Prospects for these openings should be best for applicants with a bachelor's degree and experience with a variety of programming languages and tools, including Java and other object-oriented languages.

In California, it is anticipated that employment of computer programmers will decline by 3% during 2006-2016. The median wage for this occupation in California is $76,100. According to the Advisory Committee, there is a demand for computer programmers in the San Francisco Bay area due to the growth of robotics, games, and Internet applications.

VIII. DEPARTMENTALLY APPROVED INSTRUCTIONAL MATERIALS AND EQUIPMENT

Suggested Resources — Selections from:

Java Methods A & AB: Object-Oriented Programming and Data Structures, AP Edition, Maria and Gary Litvin, Skylight Publishing, 2006

Be Prepared for the AP Computer Science Exam in Java by Maria Litvin, Skylight Publishing, 2009

Java: an Eventful Approach, Kim Bruce, Andrea Danyluk, Thomas Murtagh, Prentice Hall, 2005

Java: A Beginner's Guide, 4th Edition, Herbert Schildt, McGraw-Hill, 2006

Karel J Robot, A Gentle Introduction to the Art of Object-Oriented Programming in Java, Cafepress, 2005

Big Java, by Cay Horstman, Wiley Publishers, 2007

ICT Curriculum for AP Computer Science (on-line)

Web resources:

http://apcentral.collegeboard.com/apc/public/courses/teachers_corner/4483.html (Home page for AP Computer Science on College Board Web site)

http://apcentral.collegeboard.com/apc/public/repository/ap-computer-science-course-description.pdf (Course Description for AP Computer Science)

http://apcentral.collegeboard.com/apc/members/repository/ap07_compsci_teachersguide.pdf (Teacher's Guide for AP Computer Science)

http://apcentral.collegeboard.com/apc/members/courses/teachers_corner/50030.html ("Nifty Assignments" for AP Computer Science

Java IDE's:

www.jetbrains.com/idea/

www.eclipse.org/downloads/ (Open source)

www.netbeans.org/ (Open source)

www.bluej.org (BlueJ)

Equipment: Fully equipped computer lab and high speed Internet connection

APPENDIX

Download the following documents from AP Central at the College Board Web site:

AP Computer Science Home Page
http://apcentral.collegeboard.com/apc/public/courses/teachers_corner/4483.html

AP Computer Science-A Course Description:
http://apcentral.collegeboard.com/apc/public/repository/ap-computer-science-course-description.pdf

AP Computer Science Java Subset – appendix A of above Course Description
http://apcentral.collegeboard.com/apc/public/repository/ap-computer-science-course-description.pdf

Standard Java Library Methods Required for AP Computer Science A – appendix B of above Course Description
http://apcentral.collegeboard.com/apc/public/repository/ap-computer-science-course-description.pdf

AP Computer Science Teacher's Guide:
http://apcentral.collegeboard.com/apc/members/repository/ap07_compsci_teachersguide.pdf

Student Activities – appendix B of above Teacher's Guide
http://apcentral.collegeboard.com/apc/members/repository/ap07_compsci_teachersguide.pdf

Nifty Assignments
http://apcentral.collegeboard.com/apc/members/courses/teachers_corner/50030.html

GridWorld Case Study
http://apcentral.collegeboard.com/apc/public/courses/teachers_corner/151155.html