# Student Programs



Quality career training that works!

Approved by California Department of Education Accredited by Western Association of Schools and Colleges UC certified "a-g"



# Information Technology Industry Sector

# **Computer Programming/Computer Science**

Curriculum

Approved by Contra Costa County Board of Education June 1, 2000 Revised and approved by Contra Costa County Board of Education February 7, 2007 California Career Technical Education Standards Included Revised October 26, 2009

www.cocoschools.org/rop

#### CONTRA COSTA COUNTY OFFICE OF EDUCATION

Contra Costa County Superintendent of Schools Joseph A. Ovick, Ed.D.

Associate Superintendent Student Programs & Services Karen Sakata

Student Programs Directors Marie McClaskey Jane Shamieh, J.D.

Contra Costa County Board of Education Daniel L. Borsuk Michèle L. Foster, Ed.D. Daniel A. Gomes Pamela M. Mirabella Glenn A. Ruley

**ROP Computer Science Curriculum Committee** 

Lindi Huntsman—Chair Bhupinder Anwar Carol Cho Donald Griffiths Judith McCall Dirk Sikkema

A Publication of the Contra Costa County Office of Education 77 Santa Barbara Road Pleasant Hill, CA 94523(925) 942-3368

#### **ROP Computer Programming/Computer Science**

(AP Computer Science A – Java)

#### PHILOSOPHY

ROP delivers state-of-the-art educational programs to high school and adult students to prepare them for challenging careers, higher education and lifelong learning.

#### **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	ROP 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
  - o Develop appropriate listening, speaking, and presentation skills
  - Use technology to enhance communication
  - o Read and comprehend industry-related material
  - o 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
  - o Demonstrate appropriate business ethics and etiquette
  - o 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
  - o Demonstrate problem-solving, computational, and research skills
- 4. Work productively both as individuals and as *team members*.
  - o Demonstrate initiative and resourcefulness
  - o Brainstorm and collaborate with others
  - o Demonstrate the ability to assume a leadership role
  - o Give and receive constructive feedback

#### III. COURSE OUTLINE

	Course Outline	Career Technical Education Standards		Suggested Activities/Assessment
1.	Orientation (10 hrs)	Foundation Standards	0	Orientation lecture
Α.	Classroom procedures and safety unit	6.0 Health and Safety	0	Lecture and demonstration of safety material
В.	<ol> <li>Review of safety and guidelines</li> <li>Ergonomics and computer use</li> <li>Overview of the field of computer science and computer programming</li> </ol>	<ul> <li>Students understand health and safety policies, procedures, regulations, and practices, including the use of equipment and handling of hazardous materials:</li> <li>Understand the environmental and ergonomic risks associated with the use of business equipment and the financial impact of an unsafe work environment.</li> <li>Technology <ul> <li>Students know how to use contemporary and emerging technological resources in diverse and changing personal, community, and workplace environments:</li> </ul> </li> <li>Understand the use of technological resources to gain access to, manipulate, and produce information, products, and services.</li> </ul>	0	Lecture on the history and evolution of computer programming languages
		4.5 Know procedures for maintaining secure information,		

Course Outline			Career Technical Education Standards		Suggested Activities/Assessment		
			preventing loss, and reducing risk.				
2	Object	t-oriented program design (30 hrs)	Pathway Standards	0	Lecture, demonstration, and class discussion:		
A.	Progra	am design	D3.0 Students understand the creation and design of a		The overall goal for designing a piece of		
	1.	Read and understand a problem	software program:		software (a computer program) is to correctly		
		description, purpose, and goals	D3.1 Analyze customers' needs and requirements for		solve the given problem. At the same time, this		
	2.	Apply data abstraction and encapsulation	software.		designing a program that is understandable		
	3.	Read and understand class specifications and relationships among the classes	D3.2 Know how specifications and codes are developed for new and existing software applications.		can be adapted to changing circumstances, and has the potential to be reused in whole or in		
		a. "is-a" relationship	D3.3 Understand the abstract organization of information		part. The design process needs to be based on		
		b. "has-a" relationship	and how programs maintain the properties of the data	ć	a thorough understanding of the problem to be		
	4.	Understand and implement a given class	structure while they perform such operations as search, insert, or load-balancing		Solved.		
	Б	Identify rousable companents from	D3.4 Know multiple ways in which to store retrieve and	0	design. This includes an understanding of how		
	5.	existing code using classes and class	access information.	1	to apply data abstractions (classes and arrays).		
		libraries	D3.5 Understand how to track software versions.	0	Describe the inheritance and composition		
В.	Class	design	Foundation Standards	I	relationships among the different classes that		
	1.	Design and implement a class	2.0 Communications	(	comprise a program.		
	2.	Choose appropriate data representation and algorithms	Students understand the principles of effective oral, written, and multimedia communication in a variety of	0	given the specifications for the classes		
	3.	Apply functional decomposition	formats and contexts.		Involved—Which classes are subclasses of other classes		
	4.	Extend a given class using inheritance	2.1 Reading		Given a design for a class, either their own or		
C.	Advan	ced topics — optional	(2.4) Synthesize the content from several sources or works		one provided, students should then be able to		
	1.	Specify the purpose and goals for a problem	by a single author dealing with a single issue; paraphrase the ideas and connect them to other	i	implement the class; extend a given class using inheritance, thereby creating a subclass with		
	2.	Decompose a problem into classes	sources and related topics to demonstrate	1	modified or additional functionality.		
		a. Define relationships	comprehension.	0	Write a class that implements an interface.		
		b. Define responsibilities	(2.6) Demonstrate use of sophisticated learning tools by	0	Design a program to develop a solution that		
	3.	Design and implement a set of interacting	tollowing technical directions (e.g., those found with graphic calculators and specialized software	i	includes the following:		
	Λ	Classes	programs and in access guides to World Wide Web	-	Appropriate use of inheritance from another class using the keyword "extends"		
	4. 5	Choose appropriate advanced data	sites on the Internet).		Appropriate implementation of an interface		
	J.	structures and algorithms			using the keyword "implements"		
		-			Declaration of constructors and methods		

Course Outline			ourse Outline	C	Career Technical Education Standards		Suggested Activities/Assessment
3.	Progr	ram imple	ementation (30 hrs)	Pathw	ay Standards	0	Lecture, demonstration, class discussion:
Α.	Imple	mentation	1 techniques	D1.0	Students understand the strategies necessary		Classes that fill common needs should be built
	1.	Metho	odology		to define and analyze systems and software		so they can be reused easily in other programs.
		а.	Object-oriented development		requirements:		Object-oriented design is an important part of program implementation
		b.	Top-down development	D1.1	Develop information technology-based strategies		Find an implementation.
			c. Encapsulation		and project plans to solve specific problems.	0	in an object-oriented approach to programming
			and information hiding	D1.3	Know the effective use of tools for software		including console-based character I/O,
		d.	Procedural abstraction		development.		graphical user interfaces, and applets.
В.	Progra	amming c	constructs	D1.4	Know the software development process.	0	Demonstrate an understanding of the concept
	1.	Primiti	ive types vs. objects	D2.0	Students understand programming languages:		of recursion and to trace recursive method calls
	2.	Declar	ration	D2.1	Know the fundamentals of programming languages	0	Recognize the possibilities of reusing
		а.	Constant declarations		and concepts.		components of one's own code or other
		b.	Variable declarations	D2.2	Compare programs by using control structures,		examples of code in different programs.
		C.	Class declarations		procedures, functions, parameters, variables, error	0	See attached Appendix A "AP Computer
		d.	Interface declarations	5.0.0	recovery, and recursion.		Science Java Subsel"
		e.	Method declarations	D2.3	Understand digital logic, machine-level representa-	0	Library Methods Required for AP Computer
		f.	Parameter declarations		of assembly level programming architecture		Science A"
	3.	Consc	ole output	2.2	Writing	0	Activity: <i>Recursion Worksheets</i> (see Appendix
		а.	System.out.print/println	2.2	Write technical decuments:	Ŭ	B of <i>Teacher's Guide</i> )
	4.	Contro	ol	2.0		0	Activity: AP Computer Science Project I (see
		а.	Methods		a. Report information and convey ideas		Appendix B of Teacher's Guide)
		b.	Sequential		b Offer detailed and accurate encoifications	0	Programming Assignment: Hailstone Sequence;
		C.	Conditional		b. Otter detailed and accurate specifications.		download from "Nifty Assignments" at AP Central
		d.	Iteration		c. Include scenarios, definitions, and	0	Programming Assignment: Cat and Mouse;
			e.		troubleshooting guide)		download from "Nifty Assignments" at AP
			Understand and		troubleshooting guide).		Central
			evaluate recursive				
C	lovo '	libron do	methous				
С.	Java I	inn ar à cla	SSUS				
	• 5	see allach	ieu Appendix A and B				
						1	

	Course Outline	Career Technical Education Standards	Suggested Activities/Assessment		
4.	Program analysis (30 hrs)	Pathway Standards	o Lecture, demonstration, and class discussion:		
A.	<ol> <li>Testing</li> <li>Test classes and libraries in isolation</li> <li>Identify boundary cases and generate appropriate test data</li> </ol>	<ul> <li>D4.0 Students understand the process of testing, debugging, and maintaining programs to meet specifications:</li> <li>D4.1 Know the steps involved in the software-testing process</li> </ul>	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		
В.	<ul> <li>3. Perform integration testing</li> <li>Debugging</li> <li>1. Categorize errors: <ul> <li>a. Compile-time</li> <li>b. Run-time</li> <li>c. Logic</li> </ul> </li> <li>2. Identify and correct errors</li> <li>3. Employ techniques such as: <ul> <li>a. Using a debugger</li> <li>b. Adding extra output statements</li> </ul> </li> </ul>	<ul> <li>D4.2 Know the methodologies of program maintenance to preserve intended program applications and the operation of scheduled batch jobs and real-time jobs.</li> <li>1.2 Science—Investigation and Experimentation</li> <li>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.</li> <li>1.d Formulate explanations by using logic and evidence.</li> </ul>	<ul> <li>requirements when applied to different data sets.</li> <li>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.</li> <li>Make informal comparisons of running times of different pieces of code by counting the number of loop iterations needed for a computation.</li> </ul>		
C. D. F. G.	<ul> <li>Understand and modify existing code</li> <li>Extend existing code using inheritance</li> <li>Understand error handling</li> <li>1. Understand runtime exceptions</li> <li>Reason about programs</li> <li>1. Pre- and post-conditions</li> <li>2. Assertions</li> <li>Analysis of algorithms</li> <li>1. Informal comparisons of running times</li> <li>2. Exact calculation of statement exception</li> </ul>	<ul> <li>2.2 Writing</li> <li>2.3 Write expository compositions, including analytical essays and research reports: <ul> <li>a. Marshal evidence in support of a thesis and related claims, including information on all relevant perspectives.</li> <li>b. Convey information and ideas from primary and secondary sources accurately and coherently.</li> <li>c. Make distinctions between the relative value and significance of specific data, facts, and ideas.</li> </ul> </li> </ul>	<ul> <li>Write a report on a case study</li> <li>Case Study of Marine Biology Simulation, (download from "Nifty Assignments" at AP Central College Board Web site.)</li> <li>Case study of GridWorld (download files from the AP Central College Board Web site)</li> <li>See current AP Computer Science Case Study (download from AP Central) for activities such as the following:</li> <li>modifying the procedural and data organization of the case study program to correspond to changes in the program</li> </ul>		
H.	Numerical representations and limits1.Representations of numbers in different bases	a. Include visual alds by employing appropriate technology to organize and record information on charts, maps, and graphs.	<ul> <li>extending the case study program by writing new code (including new methods for existing classes, new subclasses extending existing classes, and new</li> </ul>		

	Course Outline	Career Technical Education Standards	Suggested Activities/Assessment
I.	<ol> <li>Limitations of finite representations         <ul> <li>Integer bounds</li> <li>Imprecision of floating-point representations</li> <li>Round-off error</li> </ul> </li> <li>Advanced topics – optional</li> <li>Throw runtime exceptions</li> <li>Big-Oh notation</li> <li>Worst-case and average-case time and space analysis</li> </ol>		<ul> <li>classes)</li> <li>evaluating alternatives in the representation and design of objects and classes</li> <li>evaluating alternative incremental development strategies</li> <li>understanding how the objects/classes of the program interact.</li> </ul>
5. A. B. C. D. E.	Standard data structures (30 hrs)Simple data types1.Int2.Boolean3.DoubleClassesListsArrays1.One-dimensional arrays2.Two-dimensional arrays2.Two-dimensional arraysAdvanced topics – optional1.Linked lists (singly, doubly, circular)2.Stacks3.Queues4.Trees5.Heaps6.Priority queues7.Sets8.Maps	<ul> <li>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:</li> <li>5.1 Apply appropriate problem-solving strategies and critical thinking skills to work-related issues and tasks.</li> <li>5.2 Understand the systematic problem-solving models that incorporate input, process, outcome, and feedback components.</li> <li>5.3 Use critical thinking skills to make informed decisions and solve problems.</li> </ul>	<ul> <li>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.</li> <li>Use the standard representations of integers, real numbers, and Boolean (logical) variables.</li> <li>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 <i>Course Description</i>, download from AP Central)</li> <li>Activity: <i>The Mathematical Matrix Class</i> (See appendix B of <i>Teacher's Guide</i>, download at AP Central)</li> <li>Programming Assignment: <i>Car Rental</i> (download from "Nifty Assignments" at AP Central)</li> <li>Programming Assignment: <i>Twelve Days of Christmas</i> (download from "Nifty Assignments" at AP Central)</li> </ul>

	Course Outline	Career Technical Education Standards			Suggested Activities/Assessment
6.	Standard algorithms (30 hrs)	D5.0 Students u	nderstand the importance of quality	0	Lecture, demonstration, and class discussion:
А.	Operations on data structures previously listed	assurance efficient pr	tasks in producing effective and oducts:		Standard algorithms serve as examples of good solutions to standard problems. Many are
	2. Insertions	D5.1 Know the st	andards and requirements for software		intertwined with standard data structures. These algorithms provide examples for analysis of
B.	3. Deletions Searching	D5.2 Know comm	non quality assurance tasks and their	0	program efficiency. Describe standard algorithms for accessing
	1. Sequential	D5.3 Understand	the ways in which specification		arrays, including traversing an array and inserting into and deleting from an array.
C.	Sorting	the modifica	ation of programs.	0	Explain the two standard searches, sequential search and binary search, and the relative
	1. Selection	D5.4 Know variou	us sorting and searching methods and rative advantages		efficiency of each.
	<ol> <li>Insertion</li> <li>Mergesort</li> </ol>	D5.5 Know the ch	haracteristics of reliable, effective, and	0	Define three standard sorts that are required: the two most common quadratic sorts—
D.	Advanced topics - optional	enicient pro	uucis.		Selection sort and Insertion sort—and the efficient Merge sort.
	structures previously listed			0	Activity: <i>Searching and Sorting Worksheets</i> (see Appendix B of Teacher's Guide)
	insertions, and deletions)			0	Activity: Searching and Sorting Arrays (see Appendix B of Teacher's Guide)
	3. Quicksort			0	Programming Assignment: <i>Guessing Game</i> (download from "Nifty Assignments" at AP
	4. Heapsort				Central)
				0	Practice exams (download from AP Computer Science home page at AP Central)
				0	Algorithm Analysis Research Project (See Appendix B of Teacher's Guide)

Course Outline			Course Outline	Career Technical Education Standards			Suggested Activities/Assessment		
	7.	Comp	uting in context (10 hrs)	Pathw	ay Standards	0	Lecture, demonstration, class discussion: An		
	A.	Resport 1.	nsible use of computer systems System reliability	D6.0	Students understand the importance of effective interfaces in the interaction between humans and computer systems:		awareness of the ethical and social implications of computing systems is necessary for the study of computer science. Given the tremendous		
		2. 3.	Legal issues and intellectual property a. Licensing b. Property rights	D6.1 D6.2	Understand how to support access, privacy, and high ethical standards in computing. Use knowledge of cognitive, physical, and social interactions to create and design user-friendly		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		
	5		and copyright laws 4. Social and ethical ramifications of computer uses	Found	computer practices and applications that meet the needs of the market. lation Standards		studied in this course provide opportunities to discuss the impact of computing. Typical issues include:		
	В. С.	Major f 1. 2. 3. Svsten	nardware components Primary and secondary memory Processors Peripherals n software	<b>8.0</b> 8.2	Ethics and Legal Responsibilities Students understand professional, ethical, and legal behavior consistent with applicable laws, regulations, and organizational norms: Understand the concept and application of ethical		<ul> <li>Impact of applications using databases, particularly over the Internet, on an individual's right to privacy</li> <li>Economic and legal impact of viruses and other malicious attacks on computer</li> </ul>		
		1. 2. 3.	Language translators/compilers Virtual machines Operating systems	8.3	and legal behavior consistent with workplace standards. Understand the role of personal integrity and ethical behavior in the workplace.		<ul> <li>Need for fault-tolerant and highly reliable systems for life-critical applications and the resulting need for software engineering standards</li> </ul>		
	D.	1 ypes 1. 2	of systems Single-user systems Networks	8.5	Know how to design systems and applications to allow access to all users.		<ul> <li>Intellectual property rights of writers, musicians, and computer programmers and</li> </ul>		
		Ζ.	<ul><li>a. Understanding networking</li><li>b. Today's world is networked</li></ul>	10.1	Know how to use a variety of business- and industry-standard software and hardware, including major proprietary and open standards.	0	tair use of intellectual property Lecture on major hardware components, system software, and types of systems—how it all fits together		

	Course Outline	C	areer Technical Education Standards	Suggested Activities/Asses		
8.	Career Path (10 hrs)	3.0	Career Planning and Management	0	Discuss the personality traits that might lead to	
А.	Exploring careers in computer science and		Students understand how to make effective		success in the field of computer science	
	computer programming		decisions, use career information, and manage	0	Identify the different careers available in the	
В.	Management of time and resources	2.1	Personal career plans.		neid of computer science and computer	
C.	Resumes and cover letters	3.1	antitudes knowledge and skills necessary to		Discuss aducational requirements	
D.	Interviewing skills		succeed in careers.	0		
E.	Class discussion about desirable on-the-job characteristics, such as responsibility, flexibility, leadership, and teamwork	3.2	Understand the scope of career opportunities and know the requirements for education, training, and	0	educational experience and on-the-job experience	
		2.4	ILEIISUIE.	0	Class discussion on time management,	
		3.0	hiring process, such as job applications, résumé		responsibility, flexibility, leadership, and teamwork	
			portfolio.	0	Write resumes, cover letters and thank you	
		7.0	Responsibility and Flexibility		letters	
			Students know the behaviors associated with the demonstration of responsibility and flexibility in personal, workplace, and community settings:	0	Practice interviewing skills	
		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			
			Students understand effective leadership styles, key concepts of group dynamics, team and individual decision making, the benefits of workforce diversity, and conflict resolution:			

Course Outline	Career Technical Education Standards	Suggested Activities/Assessment
	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.	
Total hours of classroom instruction: 180		
Note: Much of the above material for items #2 - #7 of the "Co	urse Outline" and "Suggested Activities" was obtained from "Con	nputer Science A-Course Description" (May 2010-

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. ROP CERTIFICATE REQUIREMENTS

To earn 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

Intro to Comp Programming ROP.docx 10.26.09 Page 19 of 19