Scratch Your Way into (Fun) Programming

Barry Webster
bwebster@barrywebster.com
http://barrywebster.com

Michigan Computer Science Teachers Association
A Chapter OF THE Computer Science Teachers Association

Presented at MACUL Conference 2011
March 17, 2011

Scratch Your Way into (Fun) Programming

Experience ways of helping your students identify reasons computer science (CS) is important to learn, express positive attitudes towards CS, have an enjoyable introduction to programming through scratch, produce multimedia through programming, identify CS concepts, and have early and continued success with CS. Sharing is encouraged. Computers optional.

Michigan Computer Science Teachers Association

A Chapter OF THE Computer Science Teachers Association

Introductions

Michigan Computer Science Teachers Association

A Chapter OF THE Computer Science Teachers Association

Scratch Your Way into (Fun) Programming

- What is CS, What is not CS (!CS)
- Key facts about CS
- Quick visual feedback increases interest
- Introduction to Scratch
- Programming multimedia in Scratch
- Programming concepts in Scratch
- “Programming” multimedia in HTML5

Michigan Computer Science Teachers Association

A Chapter OF THE Computer Science Teachers Association

Areas of Technology in Education

- **Information technology** (IT) includes network and servers and computer hardware and software maintenance and repair.
- **Educational technology** applies technologies and methods to improve teaching and student learning.
- **Information literacy** is "the ability to know when there is a need for information, to be able to identify, locate, evaluate, and effectively use that information for the issue or problem at hand" (from National Forum on Information Literacy).
- The academic study of **Computer science**..

Michigan Computer Science Teachers Association

A Chapter OF THE Computer Science Teachers Association

What is CS? What is !CS*

*not CS

- Computer Science courses are often confused with educational technology courses (vocational courses rather than rigorous academic courses)
- Computer science is not keyboarding or using applications such as word processors, spreadsheets or databases (although important)
- Computer science is not the same as using computers to enhance learning in other academic disciplines (just having computers in the school does not teach students the fundamentals of computer science)

Computer Science

Computer Science concepts are important for the development of our children. Computer science goes beyond using technology. It goes beyond keyboarding or using applications such as word processors, spreadsheets or databases. It goes beyond using computers to enhance learning in other academic disciplines (just having computers in the school does not teach students the fundamentals of computer science). Thinking that computer science is the same as technology is like thinking that English is the same as using books, or that Science is the same as using laboratory equipment.

Michigan Computer Science Teachers Association

A Chapter OF THE Computer Science Teachers Association

What is CS?

Computer science is software and hardware design.

Michigan Computer Science Teachers Association

A Chapter OF THE Computer Science Teachers Association

What is CS?

"Computer science is the study of computers and algorithmic processes, including their principles, their hardware and software designs, their applications, and their impact on society." — The ACM Model Curriculum for K-12 Computer Science" http://csta.acm.org/Publications/sub/Documents.html

Michigan Computer Science Teachers Association

A Chapter OF THE Computer Science Teachers Association

What is CS?

Students who study computer science learn a number of vital skills that can be transferred to any subject area and contribute significantly to their performance as professionals:

- Problem solving skills: Problem definition, solution design, implementation, testing, revision; Creativity, perseverance, teamwork
- Design skills: Designing and working to specifications
- Logic and reasoning: The ability to analyze a problem and break it down into a logical sequence of steps
- Computational thinking: Drawing on fundamental concepts in computer science to analyze and solve problems; Thinking at multiple levels of abstraction

•-CSTA

Starting from Scratch

http://scratch.mit.edu/

Michigan Computer Science Teachers Association

A Chapter OF THE Computer Science Teachers Association

Exploring Scratch

http://scratch.mit.edu/

Michigan Computer Science Teachers Association

A Chapter OF THE Computer Science Teachers Association

SCRATCH
imagine • program • share

Support

How do I get started with Scratch and the Scratch website?

Visit the Getting Started Page

- Getting Started Guide
- Video Tutorials
- Scratch Cards
- Scratch Tours

Italiano
العربية
简体中文
繁體中文
Nederlands
Français
Deutsch
עברית
日本語
Polski
Español
Українська

What if I have a question about Scratch?

See the Scratch FAQ

Search the Scratch Forums

Do you have resources for educators?

See the Educators page or visit ScratchEd, an online community for educators using Scratch.

Are there resources created by Scratchers?

Check out the Scratch Wiki and the Scratch Resources site, made by and for Scratchers.

Scratch Getting Started Guide

Thursday, March 17, 2011

Programming with Scratch

Programming-with-Scratch.pdf

Most people view **computer programming** as a tedious, specialized activity, accessible only to those with advanced technical training. And, indeed, traditional programming languages like Java and C++ are very difficult for most people to learn.

Scratch, a new graphical programming language, aims to change that. Scratch takes advantage of advances in computing power and interface design to make programming more engaging and accessible for children, teens, and others who are learning to program. Key features of Scratch include:

- **Building-block programming**. To create programs in Scratch, you simply snap graphical blocks together into stacks. The blocks are designed to fit together only in ways that make syntactic sense, so there are no syntax errors. Different data types have different shapes, eliminating type mismatches. You can make changes to stacks even as programs are running, so it is easy to experiment with new ideas incrementally and iteratively.

- **Media manipulation**. With Scratch, you can create programs that control and mix graphics, animations, music, and sound. Scratch extends the media-manipulation activities that are popular in today's culture – for example, adding programmability to Photoshop-style image filtering.

Programming with Scratch

Programming-with-Scratch.pdf, cont.

- **Sharing and collaboration**. The Scratch website provides inspiration and audience: you can try out other people's projects, re- use and adapt their images and scripts, and post your own projects. The ultimate goal is to develop a shared community and culture around Scratch.

Scratch offers a **low floor** (easy to get started), **high ceiling** (ability to create complex projects), and **wide walls** (support for a wide diversity of projects). In developing Scratch, we put high priority on simplicity, sometimes even sacrificing functionality for understandability.

As students work on Scratch projects, they have opportunities to learn important **computational concepts** such as iteration, conditionals, variables, data types, events, and processes. Scratch has been used to introduce these concepts to students of many different ages, from elementary school through college. Some students transition to traditional text-based languages after getting introduced to programming with Scratch.

Scratch is built on top of the **Squeak** programming language. It was inspired by previous work on **Logo** and Squeak **Etoys**, but it aims to be simpler and more intuitive.

Scratch is an **open-source** but **closed- development** project. The source code is freely available, but the application is developed by a small team of researchers at the MIT Media Lab. -Lifelong Kindergarten Group, MIT Media Lab

ScratchReferenceGuide14.pdf

1. INTRODUCTION

version 1.4

Scratch is a new programming language that makes it easy to create interactive stories, games, and animations – and share your creations with others on the web.

This Reference Guide provides an overview of the Scratch software. If you are just getting started with Scratch, we encourage you to try the ***Getting Started Guide*** first (available from the *Support* section on the Scratch website). Then, if you want more detailed information, come back to the Reference Guide.

The Scratch website has many other resources to help you learn Scratch: Video tutorials, Scratch cards, and Frequently Asked Questions (FAQs). Please see *http://info.scratch.mit.edu/Support/*

This guide is for Scratch version 1.4, released July 2009. For the latest version of this Reference Guide, please see: *http://info.scratch.mit.edu/Support/*

http://scratched.media.mit.edu/

SCRATCH

Michigan Computer Science Teachers Association

A Chapter OF THE Computer Science Teachers Association

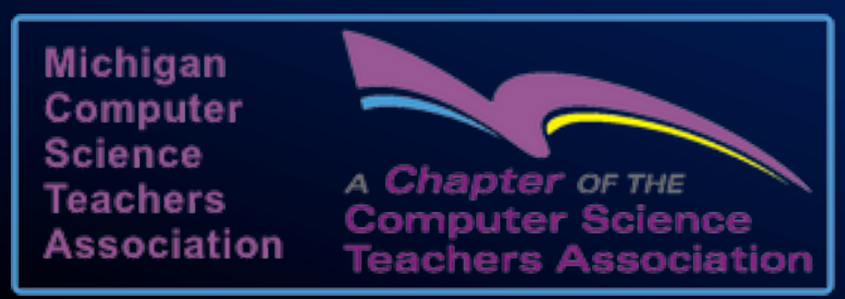
Starting from Scratch and/or Alice

http://www.alice.org/index.php

Michigan Computer Science Teachers Association

A Chapter OF THE Computer Science Teachers Association

Starting with HTML



CS Includes

- **Algorithms and Complexity**: analysis of algorithms, divide-and-conquer strategies, graph algorithms, distributed algorithms, computability theory
- **Architecture**: digital logic, digital systems, data representation, machine language, memory systems, I/O and communications, CPU design, networks, distributed computing
- **Discrete Structures**: functions, sets, relations, logic, proof, counting, graphs and trees
- **Graphics and Visual Computing**: fundamental techniques, modeling, rendering, animation, virtual reality, vision
- **Human-Computer Interaction (HCI)**: principles of HCI, building a graphical user interface (GUI), HCI aspects of multimedia, and collaboration
- **Information Management**: database systems, data modeling and the relational model, query languages, data mining, hypertext and hypermedia, digital libraries
- **Intelligent Systems**: fundamental issues, search and optimization, knowledge representation, agents, natural language processing, machine learning, planning, robotics

CS Includes

- **Net-centric Computing**: Introduction to Net- centric computing, the Web as a client-server example, network security, data compression, multimedia, mobile computing
- **Operating Systems**: concurrency, scheduling and dispatch, virtual memory, device management, security and protection, file systems, embedded systems, fault tolerance
- **Programming Fundamentals**: algorithms and problem-solving, fundamental data structures, recursion, event-driven programming
- **Programming Languages**: history and overview, virtual machines, language translation, type systems, abstraction, object- oriented (OO) programming, functional programming, translation
- **Social and Professional Issues**: ethical responsibilities, risks and liabilities, intellectual property, privacy, civil liberties, crime, economics, impact of the Internet
- **Software Engineering**: metrics, requirements, specifications, design, validation, tools, management
- ACM

Key Facts About Computer Science

•“By 2016, current government projections show that more than 800,000 high-end computing jobs will be created in the economy, making it one of the fastest growing occupational fields.

•Five of the top ten fastest growing jobs will be in computing-related fields (i.e., computer software engineer jobs expected to grow 45% over the next five to seven years).

•Computer science and computer engineering bachelor degrees are in high demand and command two of the top three average salary offers from employers among all majors.

•The percent of high schools with rigorous computer science courses fell from 40% to 27% from 2005-2009.

•The percent of high schools with introductory computer science courses fell from 78% to 65% from 2005-2009.” -http://www.csedweek.org/key-facts

Michigan Computer Science Teachers Association

A Chapter OF THE Computer Science Teachers Association

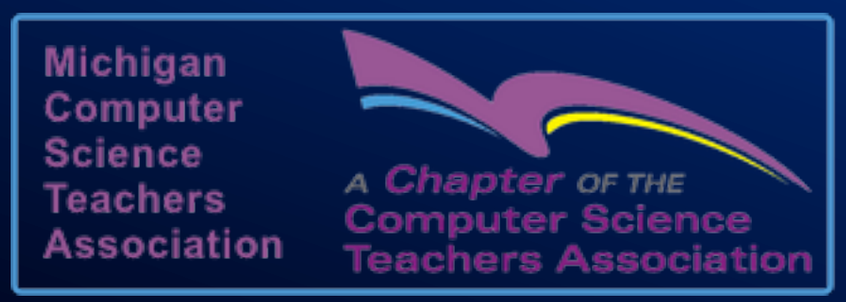
A High School Sequence

- Multimedia and Programming (sem) or
- Programming and Design for the Web
- Intermediate Programming & Data Structures (sem)
- Computer Science AB AP (year)
- Advanced Topics (sem)
- http://barrywebster.com/webster/dcdscomputer.html

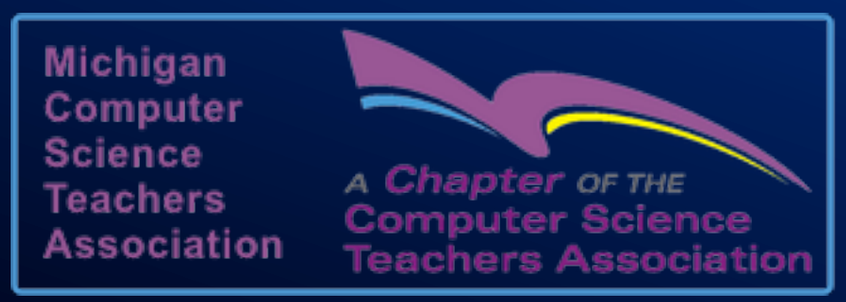
Michigan Computer Science Teachers Association

A Chapter OF THE Computer Science Teachers Association

Ideas from the Group



End


Michigan
Computer
Science
Teachers
Association
A Chapter OF THE
Computer Science
Teachers Association