



STEAM Summer Technology Program (2015) @ Chelsea School

Release Draft

Chelsea School

January 24, 2015

CONTENTS

1	Overview	1
1.1	Summary	1
1.2	Dates	1
1.3	Times	1
1.4	Location	1
1.5	Program Goals	2
1.6	Features	2
1.7	Curriculum Overview	3
1.8	Daily Schedule	3
2	Philosophy	5
2.1	Chelsea School	5
2.2	STEAM Camp	5
3	Programming Fundamentals...Program under development	7
3.1	Summary	7
3.2	Learning Objectives	7
4	Technology and the Arts [developing draft]	9
4.1	Summary	9
4.2	Prospective Activities & Topics	9
5	Cybersecurity and Digital Forensics Program	11
5.1	Description	11
5.2	Goal	11
5.3	Learning Outcomes	11
5.4	Hours	11
5.5	Prospective Scope and Sequence	11
5.6	Primary References	13
5.7	Technologies	14
5.8	Post-Secondary Paths	15
5.9	Representative Vocabulary	15
6	Game Design and App Development Program	17
6.1	Description	17
6.2	Goal	17
6.3	Learning Outcomes	17
6.4	Prospective Topics	17
6.5	Technologies	20
6.6	Resources Referenced	20

7	Program Staff [draft in progress]	21
7.1	Rik Goldman	21
7.2	Sabre Goldman	21
7.3	David Vest	21
8	Frequently Asked Questions	23
8.1	What is Chelsea School?	23
8.2	What is STEM Education? What then is STEAM education?	23
8.3	What is the location of the summer camp?	24
8.4	What are the camp hours?	24
8.5	What is the break and lunch plan?	24
8.6	How can a parent best support a participant?	24
8.7	What resources do participants need?	24
8.8	What resources are provided by the camp staff?	24
8.9	What additional resources are encouraged for participants?	25
8.10	Is homework assigned?	25
8.11	How can a participant prepare before camp starts?	25
8.12	Are there prerequisites for prospective participants?	25
9	Registration	27
9.1	Registration Options	27
9.2	Open Registration	27
9.3	Registration How-To	27
10	Recommended Reading and Viewing	29
	Index	33

OVERVIEW

1.1 Summary

Through hands-on computing, Chelsea School's STEAM¹ Summer Program provides day-long, hands-on instruction in science, technology, engineering, mathematics, & the arts for students in 4th through 12th grades.

From July 13th through the 24th, 2015, our weekdays begin at 9:00 AM with a study of relationships between culture, the arts, & innovative technologies.

Following that joint session, participants may *choose* between **Game Design & Development** or **Information Security & Forensics**.

For the remainder of the day, participants will come together for learning fundamental programming concepts by programming tools with an innovative, visual programming language from MIT; our days will end at 4:30 PM.

1.2 Dates

Weekdays between July 13th and 24th, 2015

1.3 Times

9:00 AM - 4:30 PM

1.4 Location

Chelsea School in Hyattsville, MD:

2970 Belcrest Center Dr, Suite 300 [4th Floor]

Hyattsville, MD 20782

Parking on the 4th Floor (Metro Shops Parking, above LA Fitness)

¹ Science, Technology, Engineering, Arts, and Math

In the context of this program, STEM and STEAM are used interchangeably. We feel strongly that STEM education, as it is traditionally defined, must integrate critical reading, interpretation, and critique of the arts. We note, however, that Maryland State Department of Education gives this some consideration. STEM education programs, however, do not include humanities or the arts (livescience.com offers a fairly representative definition).

From the Prince George's Plaza Metro:

1. Exit Metro Station and Cross Belcrest Center Drive
2. Enter Ground-Floor Door marked 'Chelsea School/LA Fitness'
3. Take Elevator to 4th Floor and Ring Buzzer

1.5 Program Goals

- Promote literacy
- Practice project management strategies
- Encourage professional collaboration and consultation
- Explore post-secondary programs, careers, and professional certification options
- Discover programmatic solutions to common problems
- Promote digital literacy
- Practice mind-mapping and other discovery techniques
- Introduce sound and valid logic
- Promote creative problem solving and critical thinking
- Provoke innovation and invention
- Encourage self-advocacy
- Encourage life-long learning

1.6 Features

1.6.1 Metro Accessible

Our campus is conveniently located across the street from the Prince George's Metro stop.

1.6.2 State of the Art Facilities

Our new, state-of-the-art facility features interactive whiteboards; gigabit ethernet; & over 100 student workstations equipped for equitable access to appropriate assistive and instructional technologies.

1.6.3 Research-based Instruction

Grounded in [Chelsea School's philosophy, mission, and values](#), and supported by highly experienced instructors and technologists, participants will receive *individualized* instruction; they will practice reading across the curriculum; and will take part in post-secondary exploration & preparation.

1.7 Curriculum Overview

1.7.1 Programming Concepts

This game-based introduction to the logic underlying software engineering focuses on coding fundamentals using visual & text-based interpreted languages & their syntaxes; in addition, this session prepares participants for either our **Game Design and Development** or **Information Security & Digital Forensics** sessions.

1.7.2 Game Design & Development

Participants will have hands-on experience building a game in professional game development engines. Using 3D models & code produced in C# & Javascript, they will assemble the terrain, objects & other elements of the game, capture input from users & script game physics & AI behavior.

1.7.3 Cybersecurity & Digital Forensics

Attendees will learn & apply basic concepts of programming, computer forensics, cryptography, steganography, system vulnerability assessment, & project management from a series of gaming & simulation activities & hands-on exercises.

1.7.4 Arts & Technology

Participants will explore the role of arts & creativity in technology, including the ways in which literature, visual arts, & music influence — and are influenced by — science & technology. Participants will engage in hands-on projects including creative writing workshops, crafts & more.

1.8 Daily Schedule

Weekdays

- Arts and Technology
- Break
- Independent Projects
- Lunch
- Programming Fundamentals
- Breakout Session: **Game Design** *or* **Infosec and Forensics**

PHILOSOPHY

2.1 Chelsea School

2.1.1 Mission

Chelsea School educates promising students with specific language-based learning differences in a rigorous, individualized college preparatory environment to become lifelong, independent learners.

2.1.2 Vision

A model educational community where each student is valued, has equitable access to opportunity, embraces their differences and celebrates their achievements.

2.1.3 Values

- Teach literacy across content areas
- Program based on each student's individual needs
- Utilize innovative technologies
- Develop the full potential of every student
- Foster a respectful, collaborative, and nurturing community
- Affirm and celebrate diversity
- Hold each other to an expectation of personal responsibility, integrity and excellence

2.2 STEAM Camp

2.2.1 Values

- Equality
- Social and economic justice
- Differentiated Instruction
- Constructivist and Constructionist Pedagogy

- Connected Learning ¹
- Authentic Learning and Assessment
- Multi-Modal Learning and Instruction
- Rigorous relevant material
- Independent practice
- Risk-taking

2.2.2 Rational

[draft in progress]

- For every job posting for a bachelor's degree recipient in a non-STEM field there were 5 entry level postings for a bachelor's degree recipient in a STEM field. ([STEMConnector](#))
- By 2018, the very significant bulk of STEM careers will be in the field of computing: 71% (traditional engineering is second at 16%). ([STEMConnector](#))
- Only 10% of students in the United States receive credits for computing related courses in elementary, primary, or secondary school courses. ([Code.org](#))
- Computing careers offer very broad variety of entry-level position requirements, allowing for significant personal preference when committing to a post-secondary path.

2.2.3 Critical Considerations

- Underrepresentation of women in STEM fields
- Equitable Access to resources
- Participants include reluctant readers and writers
- Participants benefit from individualized instruction well-suited to their learning styles and preferences.

¹ Connected learning is an approach to addressing inequity in education in ways geared to a networked society. It seeks to leverage the potential of digital media to expand access to learning that is socially embedded, interest-driven, and oriented toward educational, economic, or political opportunity. Connected learning is realized when a young person is able to pursue a personal interest or passion with the support of friends and caring adults, and is in turn able to link this learning and interest to academic achievement, career success or civic engagement. This model is based on evidence that the most resilient, adaptive, and effective learning involves individual interest as well as social support to overcome adversity and provide recognition (Ito). The approach knits together three crucial contexts for learning: interest-powered; peer-supported; academically oriented. In addition, it embraces these key design principles: production-centered; open networks; shared purpose.

PROGRAMMING FUNDAMENTALS...PROGRAM *UNDER DEVELOPMENT*

3.1 Summary

This game-based introduction to the logic underlying software engineering focuses on coding fundamentals using visual & text-based interpreted languages & their syntaxes & prepares participants for either our game design or cybersecurity & digital forensics sessions. Students will first be exposed to programming by creating games with Scratch, a visual language developed at MIT. Students who demonstrate mastery of the concepts introduced with Scratch will move toward programming in a high-level, interpreted, text-based language, such as Python and Ruby. Interested students will also be introduced to shell programming with bash.

3.2 Learning Objectives

The participant will:

- Demonstrate an understanding variables and variable types
- Store and recall related data in and from an array
- Differentiate between compiled and interpreted programming languages
- Articulate the difference between high and low-level programming languages
- Make appropriate use of a variety of loops
- Rely effectively on conditionals and nested conditionals to determine the behavior of a program or script
- Manage source code with a source-code management tool, such as git
- Practice pair programming in the interest of efficiency, precision, accuracy, and collaboration
- Navigate the filesystem on at least two operating systems from a graphical and command line interface
- Declare constants effectively and appropriately
- Rely on functions to reduce redundancy and discourage errors

TECHNOLOGY AND THE ARTS [*DEVELOPING DRAFT*]

4.1 Summary

Participants will explore the role of arts & creativity in technology — including the ways in which literature, visual arts, & music influence — and are influenced by — science & technology. Participants will engage in hands-on projects including creative writing workshops, reading discussions, crafts & more.

4.2 Prospective Activities & Topics

- Build a 35mm Camera
- Practice digital imaging and image manipulation with mobile devices and apps
- Create and maintain a blog
- Build a phonograph player and create your own records
- Set up a Raspberry Pi [credit-card sized computer]
- Digital image special effects
- Make electronic music with a do-it-yourself electronic synthesizer
- Create LED origami
- Animate a stuffed animal
- Create a wireless NES bluetooth controller

4.2.1 Prospective Guided & Independent Reading

- Benjamin. “Art in the Age of Mechanical Reproduction.”
- Bueno. *Lauren Ipsum*
- Danielewski. *House of Leaves* (excerpts).

CYBERSECURITY AND DIGITAL FORENSICS PROGRAM

5.1 Description

The *Cybersecurity and Digital Forensics* course is a component of a two-week summer program at Chelsea School in Hyattsville, Maryland, for middle and secondary-school students who are interested in the growing field of cybersecurity¹.

This ten-day experience provides hands-on activities focused on computing and cybersecurity topics. Attendees will learn and apply basic concepts of programming, computer forensics, cryptography, steganography, system vulnerabilities and program management from a series of gaming and simulation activities and hands-on exercises.

5.2 Goal

To allow secondary students to explore and apply various topics within cybersecurity fields by linking research to practice through hands-on experience.

5.3 Learning Outcomes

To be articulated

5.4 Hours

To be defined

5.5 Prospective Scope and Sequence²

5.5.1 Cyberthreats and Motivations

- From Hacktivists to Cybercriminals, to Scriptkiddies and *lulz*

¹ In the context of this course, cybersecurity, infosec, security, network security, and information security are used interchangeably. Note too that the scope of study is limited to PC-based Microsoft and Linux-linux based systems. We do not attempt to cram mobile forensics or Macintosh server security into the program.

² A realistic plan is in place to address these topics to an appropriate degree, with time left for red-teaming and hands-on learning. As we get a snapshot of enrollment, the emphases with shift, and the delivery of content will be adjusted to meet the individual needs of participants.

5.5.2 Cybersafety

- Password best practices
- Personal disclosure
- Cyberbullying
- Getting Help or Support
- Reporting Abuse
- Generating Strong Passwords or Passphrases
- Password Management

5.5.3 Cyberethics

- Ethical Hacking
- The Ethical Hacker (Levy)
- Critical Engineering Manifesto
- System Administrator Code of Conduct & discussion of case studies/vignettes

5.5.4 Penetration Testing for Security Auditors

- Obtaining authorization from the client
- Definition of Scope with the client
- Cleaning up after yourself
- Producing a Summative Report: executive summaries and elaborated findings and recommendations
- Confidentiality and Non-Disclosure Agreements
- Legal Constraints

5.5.5 Digital Forensics

- Toolkits
- Chain of Custody
- Case Management & Reporting

5.5.6 Project Management and Professional Collaboration

- Agile Framework: Emphasis on Scrum with components of Extreme Programming
- Source code management with git (Github, Gitlab, Bitbucket)

5.5.7 Post-secondary Preparation

- Resume Development
- Career Profiles
- Post-secondary academic options
- Relevant certification options

5.5.8 Server Configuration and Administration

- Server Hardening
- Log Management
- Intrusion Detection Systems
- Network and system management

5.5.9 Threat Mitigation

- Social Engineering (exploiting human aspects of computing systems)
- Network Scanning and Enumeration
- Privilege Escalation
- Physical Access to Machines
- Malware Typology
- Brute-Force and Dictionary Attacks
- Cross-Site Scripting
- Phishing
- Session Fixation
- Session Hijacking
- SQL Injection
- Denial of Service attacks
- Known Exploits and Zero-Day Exploits

5.5.10 Cryptography and Cryptography

- Steganalysis
- Cryptoanalysis

5.6 Primary References

This session is grounded in significant research in contemporary theory, methods, and best practice as well as professional, ethical, legal conduct.

While material is drawn from myriad resources, seminal resources help inform the structure of this class:

1. *Basic Penetration Testing* (Syngress)
2. *Computer Security Literacy: Staying Safe in a Digital World* <<http://www.crcpress.com/product/isbn/9781439856185>> (CRC Press)
3. *Basic Forensics* (Syngress)
4. *Kali Linux CTF Blueprints* (Packt Pub.)
5. *Applied Network Security Monitoring* (Syngress)
6. *The Basics of Information Security, 2nd Edition* (Syngress)

5.7 Technologies

- Microsoft Windows 7
- Linux
- Apache (web server)
- MySQL (database)
- Python, Perl, PHP (interpreted scripting languages, as needed)
- Metasploit
- Backbox Linux
- Kali Linux
- Virtualization (Oracle Virtualbox, Vagrant, VMWare Player, Proxmox (hypervisor))
- Hardware firewalls
- Routers
- TCP/IP (syn, awk)
- SSL
- NMAP
- Nessus
- Mozilla Firefox security extensions
- Google Chrome security extensions
- plain text code editors (vim, gedit, emacs, notepad++)
- high level interpreted scripting languages (Python, Perl, shell scripting)
- ZendStudio
- Jira and Jira Agile (Atlassian)
- Git and Github
- metasploitable
- WPA, WEP, WPA2, WPS wireless technologies
- DD-WRT
- Linksys WRT54G Wireless Router
- Shell scripting (bash and zsh)

- IRC
- Firewall configuration (hand-on, authentic assignment): IPCOP, Barracuda

5.8 Post-Secondary Paths

5.8.1 Relevant Certification Paths

- Security+ (CompTIA)
- Network+ (CompTIA)
- Linux+ (CompTIA)
- Certified Ethical Hacker (CEH)
- ECSA: Certified Security Analyst (IACRB)
- CPT: Certified Penetration Tester (IACRB)
- Certified ScrumMaster (CSM)
- GIAC Security Essentials
- CISSP: Certified Information Systems Security Professional
- CISM: Certified Information Security Manager
- CSD: Certified Scrum Developer

5.8.2 Representative Undergrad Academic Programs

- Software Studies (UMBC)
- Network Security (Fairmont State University)
- Software Engineering (WVU)
- Digital Humanities (GMU)

5.9 Representative Vocabulary

- forensics
- cyber-
- infosec
- cybersecurity
- penetration testing
- red teaming
- server hardening
- GNU/Linux
- open-source software
- proprietary

- copyright, copyleft, and innovation, intellectual property
- case law
- case study
- imaging
- steganography
- cryptography and encryption
- handshake
- TCP/IP
- syn and ack
- pseudocode
- interpreted language
- compiled language
- high- and low-level programming languages
- chain of custody
- executive summary
- SSL/TLS
- Proxy
- Firewall
- Router
- Switch
- operating system
- filesystem
- white hat, black hat, grey hat hacking, hacktivism, scriptkiddies, cybercrime
- exploit
- CTF - security capture the flag
- security = privacy/confidentiality + data integrity + continuity of services
- intrusion detection system
- complex and strong passwords
- free-software foundation
- Electronic Frontier Foundation
- URL
- IP Address
- node

GAME DESIGN AND APP DEVELOPMENT PROGRAM

6.1 Description

Participants will be given the opportunity to design/mod their own games using the *Unity 4* game development platform. At the end of the course, students will have gained hands-on experience building a game using modern game-creation software that can be deployed as a mobile app or web-based game.

Using 3D models and code produced in C# and Javascript, participants will assemble the terrain, objects and other elements of the game, capture input from users, and script game physics and AI behavior.

6.2 Goal

Our goal for this course is to familiarize secondary-school students with the basics of game design/development from the planning stage to final product. The course will provide participants with a framework in which to experiment with the practical application of programming basics, an opportunity to learn how apps are developed and released, and the resources, encouragement, and opportunity to create in the medium of interactive narrative.

6.3 Learning Outcomes

- Game Design Basics
- C# and Javascript Coding
- Project Organization and Collaboration
- Mobile Application Deployment

6.4 Prospective Topics

6.4.1 Introduction to Gaming

- Brief History of Computer Games
- Interactive Narrative
- (Archive.org game samples)
- Introduction to Game Studies
- (Types of Games, evolution of computer gaming)

- Elements of Video Game Design
 - Art Design
 - Story Design
 - Game Mechanics, Physics, AI Behavior
- Independent v. Collaborative Development models
- Review of Terms

6.4.2 Asset Modeling and Production

- Making a Simple Mesh With Blender
- Basic Shapes
- Textures
- Animation
- Sound

6.4.3 Game Design

Assembling Tools

- Game Engines
- Languages Used
- Using a GUI Interface
- Choosing and Using a Code Editor
- Defining Your Build Goal
- Versioning (Save Your Game)

Setting the Scene

- The x,y and z planes
- Backgrounds
- Game objects
- Placement
- Lighting

Game Physics/Interaction

- Overview (Physics Best Practice)
- Movement
- Setting Boundaries For Play
- Defining an interactive element (Meshes, Coliders and Triggers)

- Randomization
- Player Input
 - Adding an Input Script
 - Syntax
 - Working with Variables
 - Timing
 - Loops
- Game Events
 - More Loops (finite loops, timed loops, while... do)
 - Scoring/Progress Tracking
 - Random Obstacles v. Artificial Intelligence
 - Simple AI

Building Different Game Types

Simple Game Styles

- The Endless Runner
- The Space Shooter (Arcade)
- The Platformer
- The Puzzle Game
- Brick Breaker and Variants

More Complex Games

- RPG and RTS Games
- Tower Defense
- Crafting Games
- 3D Shooter

6.4.4 Building Your Own Game

- Choosing a Style
- Choosing a Platform
- Choosing a Development Model
- Narrative
- Art
- Action
- GUI

6.4.5 Compiling Your Build

- Linking Scenes
- Compiling For Mobile
- Compiling for Web
- Compiling For PC
- Testing (Eclipse, Google Dev Kit, Android SDK)

6.5 Technologies

- Blender 3d modeling software (<http://www.blender.org/>)
- Unity (<http://unity3d.com/>)
- MonoDevelop (<http://www.monodevelop.com/>)
- Notepad++ (<http://notepad-plus-plus.org/>)
- GIMP (<http://www.gimp.org/>)
- Windows 7
- Google Developer Tools
- Android SDK (<http://developer.android.com/sdk/index.html>)

6.6 Resources Referenced

- <http://www.digitaltutors.com/tutorial/1764-Creating-an-Endless-Runner-Game-for-Mobile-in-Unity>
- <http://unity3d.com/learn/tutorials/projects/space-shooter>
- <http://www.informit.com/articles/article.aspx?p=2162089>
- http://en.wikibooks.org/wiki/Blender_3D:_Noob_to_Pro/Mesh_Modeling

PROGRAM STAFF [DRAFT IN PROGRESS]

7.1 Rik Goldman

7.1.1 Sessions

- Programming Fundamentals
- Infosec and Forensics

7.2 Sabre Goldman

7.2.1 Session

- Arts and Technology

7.3 David Vest

7.3.1 Sessions

- Programming Fundamentals
- Game Design and Development

FREQUENTLY ASKED QUESTIONS

8.1 What is Chelsea School?

Chelsea School is a non-profit, non-public school for students with learning differences.

Chelsea School is located in Hyattsville, MD, and serves students from the DC Metro Area.

Chelsea students are middle school and high school students (grades 5-12) with language-based learning differences and ADHD.

Chelsea faculty and staff includes dedicated teachers who are certified in special education and/or their specific content area.

The Chelsea School curriculum is designed to prepare students for the rigors of college-level courses and includes small group instruction, strategic remediation, and access to speech-language therapy, occupational therapy and counseling. (Adapted from “About Chelsea School”.)

8.2 What is STEM Education? What then is STEAM education?

This curriculum seeks to merge the boundaries of science, technology engineering and math (STEM), while explicitly integrating these subjects with arts and the humanities.

Maryland State Department of Education gives this some consideration in its *Maryland State STEM Standards of Practice Framework*¹.

¹ From *Maryland State STEM Standards of Practice Framework* (2012):

STEM education is an approach to teaching and learning that integrates the content and skills of science, technology, engineering, and mathematics. STEM Standards of Practice guide STEM instruction by defining the combination of behaviors, integrated with STEM content, which are expected of a proficient STEM student. These behaviors include engagement in inquiry, logical reasoning, collaboration, and investigation. The goal of STEM education is to prepare students for post-secondary study and the 21st century workforce.

STEM education removes the artificial barriers that isolate content and allows for an integrated instructional approach. The curriculum should allow students to develop life skills and apply content knowledge within a real world context. STEM education is active and focuses on a student-centered learning environment. Students engage in questioning, problem solving, collaboration, and hands-on activities while they address real life issues. In STEM education, teachers function as classroom facilitators. They guide students through the problem-solving process and plan projects that lead to mastery of content and STEM proficiency. STEM proficient students are able to answer complex questions, investigate global issues, and develop solutions for challenges and real world problems while applying the rigor of science, technology, engineering, and mathematics content in a seamless fashion. STEM proficient students are logical thinkers, effective communicators and are technologically, scientifically, and mathematically literate. (4)

There are two goals for STEM education in high school. The first goal is on the development of STEM proficient students. All students will continue to grow in their STEM proficiency as they progress from grades 9-12. Students demonstrate independence and become more focused and sophisticated in their approach to answering complex questions, investigating global issues, and developing solutions for challenges and real world problems. STEM proficient students graduate with the basic skills and knowledge required to pursue post-secondary study or work in any field.

The second goal for STEM education in high school is on the advanced preparation of students for post-secondary study and careers in science, technology, engineering, or mathematics. High school provides a unique opportunity for students to explore different career paths and college majors through advanced coursework, career academies, magnet programs, STEM academies, specialized STEM programs, internships, and dual enrollment opportunities. Specific programs to address the needs for advanced preparation of students shall be determined by individual schools systems. (5)

However, we feel strongly that

1. Computing is a core literacy
2. The ability to read, interpret, and critique cultural productions is a necessary condition of digital literacy and computing.

Therefore, we have worked toward adopting the acronym *STEAM* and the phrases *STEAM Education* and *STEAM Summer Program*.

8.3 What is the location of the summer camp?

The campus of Chelsea School in Hyattsville, MD: see <http://chelseaschool.edu/about/directions> for elaboration.

8.4 What are the camp hours?

Weekdays from 9:00 AM to 4:30 PM from July 13th through July 24th.

8.5 What is the break and lunch plan?

Brown bag lunches are recommended for participants. Lunch is scheduled between morning and afternoon sessions. 15 minute breaks are provided between sessions.

8.6 How can a parent best support a participant?

Engage in substantive discussion and enquiry about each day's content; provide an opportunity and support (perhaps working as a pair) for independent enquiry and practice after each day's programs conclude.

8.7 What resources do participants need?

Access to a computer and internet are recommended, but not necessary.

8.8 What resources are provided by the camp staff?

All content materials, including e-texts, opportunities to screen films and videos; materials for arts and crafting; Windows 7 and GNU/Linux workstations. For the protection of the population, internet content filters are in place; when legitimate content is incidentally or mistakenly filtered, we will act quickly to make it available.

In addition, participants will receive one USB drive containing course content and recommended software. Each participant will receive a second USB drive containing a live operating system (allows participants to explore alternative operating systems or functional toolkits without effecting home computers in any way).

8.9 What additional resources are encouraged for participants?

Ideal conditions include home access to an Intel i5 or AMD equivalent PC with 4-8 GB RAM and the ability to boot from USB drives; unfettered access to the Internet; a reading partner or guide; curiosity and the courage to pursue independent research related to program content.

8.10 Is homework assigned?

Individualized and independent practice is recommended for each participant; opportunities for daily practice, with support, will be provided during the morning “Technology and the Arts” course.

8.11 How can a participant prepare before camp starts?

Our first recommendation is to look into the “Recommended Reading and Viewing” and looking into texts that provoke interest. Our second recommendation is to keep a reading log of texts, videos, and webpages that are read or scanned in anticipation of this summer program.

We encourage eager participants to introduce themselves and ask for suggested starting points based on their interests and program paths: Initiate contact with the [contact form](https://secure.jotformpro.com/form/43524617695968) at <https://secure.jotformpro.com/form/43524617695968>.

8.12 Are there prerequisites for prospective participants?

There is a grade level restriction: We welcome participants going in to grades 4 through 12.

REGISTRATION

9.1 Registration Options

9.2 Open Registration

9.3 Registration How-To

RECOMMENDED READING AND VIEWING

What follows is a partial list of recommended texts for reading and viewing for our STEAM Summer Program participants and instructors. This list includes relevant fiction and non-fiction, for viewing, reading, and e-reading. Note that many of the resources listed here, such as the *BBS* Documentary and Cory Doctorow and Richard M. Stallman are freely available on the Web.

The Digital Manifesto Archive. Web. 24 Jan. 2015. <<https://digitalmanifesto.omeka.net/>>.

A Cypherpunk's Manifesto. Web. 30 Dec. 2014. <<http://www.activism.net/cypherpunk/manifesto.html>>.

Beyer, Kurt. *Grace Hopper and the Invention of the Information Age*. Cambridge, Mass.: MIT, 2009. Print.

Blake, Victoria. *Cyberpunk: Stories of Hardware, Software, Wetware, Evolution and Revolution*. Or.: Underland, 2013. Print.

Briggs, Jason. *Python for Kids: A Guide for Beginners*. San Francisco, Calif.: No Starch :, 2012. Print.

War Games. United Artists, 1983. Film.

Bueno, Carlos. *Lauren Ipsum: A Story about Computer Science and Other Improbable Things*. S.I.: O'Reilly Media, 2014. Print.

Carroll, John. *Agile Project Management in Easy Steps*. Warwickshire, United Kingdom: In Easy Steps, 2012. Print.

Carrott, James H., and Brian David Johnson. *Vintage Tomorrows: A Historian and a Futurist Journey through Steam-punk into the Future of Technology*. Farnham: O'Reilly, 2013. Print.

Cline, Ernest. *Ready Player One*. New York: Crown, 2011. Print.

Coleman, E. Gabriella. *Coding Freedom: The Ethics and Aesthetics of Hacking*. Princeton: Princeton UP, 2013. Print.

Coleman, Gabriella. *Hacker Culture and Politics Syllabus*. Web. 24 Jan. 2015. <<http://gabriellacoleman.org/wp-content/uploads/2012/09/Coleman-Hacker-syllabus-2012-FINAL.pdf>>.

Coleman, Gabriella. *Hacker, Hoaxer, Whistleblower, Spy: The Story of Anonymous*. S.I.: Verso, 2014. Print.

Doctorow, Cory, and Amanda Palmer. *Information Doesn't Want to Be Free: Laws for the Internet Age*. Print.

Doctorow, Cory. *Down and out in the Magic Kingdom*. New York: Tor, 2003. Print.

Doctorow, Cory. *Eastern Standard Tribe*. New York: Tor, 2004. Print.

Doctorow, Cory. "When Sysadmins Ruled the World." *Overclocked: Stories of the Future Present*. New York: Thunder's Mouth :, 2007. Print.

Doctorow, Cory. *Little Brother*. New York: Tom Doherty Associates, 2008. Print.

Doctorow, Cory. *For the Win*. New York: Tor, 2010. Print.

Doctorow, Cory. *Pirate Cinema*. New York: Tor Teen, 2012. Print.

- Doctorow, Cory, and Charles Stross. *Rapture of the Nerds*. New York: Tor, 2012. Print.
- Doctorow, Cory. *Pirate Cinema*. New York: Tor Teen, 2012. Print.
- Engelbreton, Pat. *The Basics of Hacking and Penetration Testing Ethical Hacking and Penetration Testing Made Easy*. 2nd ed. Amsterdam: Syngress, an Imprint of Elsevier, 2013. Print.
- Extreme Programming Pocket Guide*. Beijing: O'Reilly, 2003. Print.
- Ford, Jerry Lee. *Scratch Programming for Teens*. Boston, MA: Course Technology, 2009. Print.
- Gibson, William, and Jack Womack. *Neuromancer*. New York: Ace, 2000. Print.
- Goldstein, Emmanuel. *The Best of 2600: A Hacker Odyssey*. Indianapolis, IN: Wiley Pub., 2008. Print.
- Hafner, Katie, and Matthew Lyon. *Where Wizards Stay up Late: The Origins of the Internet*. New York: Simon & Schuster, 1996. Print.
- Hurley, Chris. *Penetration Tester's Open Source Toolkit*. Burlington, MA.: Syngress Pub., 2007. Print.
- Kubica, Jeremy. *Computational Fairy Tales*. North Charleston, S.C.: CreateSpace], 2012. Print.
- Langr, Jeff, and Tim Ottinger. *Agile in a Flash: Speed-learning Agile Software Development : Agile Cards for Agile Teams*. Raleigh: Pragmatic help, 2011. Print.
- Latterell, Catherine G. *ReMix: Reading + Composing Culture*. 2nd ed. Boston, Mass.: Bedford/St. Martin's, 2010. Print.
- Lessig, Lawrence. *Code*. Version 2.0. ed. New York: Basic, 2008. Print.
- Lessig, Lawrence. *Free Culture: How Big Media Uses Technology and the Law to Lock down Culture and Control Creativity*. New York: Penguin, 2004. Print.
- Lessig, Lawrence. *Remix*. Bloomsbury Academic, 2008. Print.
- Levy, Steven. *Hackers: Heroes of the Computer Revolution: 25th Anniversary Edition*. Sebastopol, CA: O'Reilly, 2010. Print.
- McGrath, Mike. *Linux in Easy Steps*. 4th ed. Southam, Warwickshire, UK: In Easy Steps/Computer Step, 2008. Print.
- McManus, Sean. *Scratch Programming in Easy Steps*. Leamington Spa, Warwickshire, U. K.: In Easy Steps, 2013. Print.
- Minsky, Marvin Lee. *The Society of Mind*. New York: Simon and Schuster, 1986. Print.
- Moody, Glyn. *Rebel Code: The inside Story of Linux and the Open Source Revolution*. Cambridge, Mass.: Perseus Pub., 2001. Print.
- Muniz, Joseph, and Aamir Lakhani. *Web Penetration Testing with Kali Linux: a Practical Guide to Implementing Penetration Testing Strategies on Websites, Web Applications, and Standard Web Protocols with Kali Linux*. Birmingham: Packt, 2013. Print.
- Poulsen, Tim. *CompTIA Linux Certification Powered by LPI*. Fairport, NY: Axzo, 2011. Print.
- Rao, Umesh Hodeghatta, and Umesha Nyak. *The InfoSec Handbook: an Introduction to Information Security*. New York: Friends of ED/ApressOpen, 2014. Print.
- Raymond, Eric S. How To Become A Hacker. Web. 24 Jan. 2015. <<http://www.catb.org/esr/faqs/hacker-howto.html>>.
- Raymond, Eric S. *The Cathedral and the Bazaar Musings on Linux and Open Source by an Accidental Revolutionary*. Sebastopol: O'Reilly Media, 2001. Print.
- Regalado, Daniel. *Gray Hat Hacking: The Ethical Hacker's Handbook*. 4th ed. S.l.: McGraw-Hill, 2014. Print.
- Rushkoff, Douglas. *Program or Be Programmed: Ten Commands for a Digital Age*. Berkeley, CA: Soft Skull, 2011. Print.

- Sammons, John. *The Basics of Digital Forensics the Primer for Getting Started in Digital Forensics*. Waltham, MA: Syngress, 2012. Print.
- Sande, Warren, and Carter Sande. *Hello World!: Computer Programming for Kids and Other Beginners*. Greenwich, Conn.: Manning, 2009. Print.
- BBS [Documentary]. 2005. Film.
- Hackers*. MGM/UA Home Video, 1998. Film.
- Sousa, David A., and Thomas Pilecki. *From STEM to STEAM: Using Brain-compatible Strategies to Integrate the Arts*. Print.
- Stallman, Richard. *Free Software, Free Society: Selected Essays of Richard M. Stallman*. Boston, MA: Free Software Foundation, 2002. Print.
- Stephenson, Neal. *Snow Crash*. New York: Bantam, 1992. Print.
- Stephenson, Neal. *The Diamond Age, Or, Young Lady's Illustrated Primer*. New York: Bantam, 1995. Print.
- Stephenson, Neal. *In the Beginning ... Was the Command Line*. New York: Avon, 1999. Print.
- Super Scratch Programming Adventure! Learn to Program by Making Cool Games!* San Francisco: No Starch, 2012. Print.
- Super Scratch Programming Adventure!: Learn to Program by Making Cool Games* (covers Scratch 2.0). Print.
- "The Critical Engineering Manifesto." The Critical Engineering Manifesto. Web. 30 Dec. 2014. <<http://criticalengineering.org/>>.
- "The Scrum Guide." Scrum Guide. Web. 30 Dec. 2014. <<http://www.scrumguides.org/scrum-guide.html>>.
- Thomas, Douglas. *Hacker Culture*. Minneapolis: U of Minnesota, 2002. Print.
- Thorn, Alan. *Unity 4 Fundamentals Get Started at Making Games with Unity*. Burlington, MA: Focal, 2014. Print.
- Turkle, Sherry. *Life on the Screen: Identity in the Age of the Internet*. New York: Simon & Schuster, 1995. Print.
- Walker, Matthew. *CEH, Certified Ethical Hacker: Exam Guide : All-in-one*. New York: McGraw-Hill, 2012. Print.
- Wark, McKenzie. *A Hacker Manifesto*. Cambridge, MA: Harvard UP, 2004. Print.
- Wark, McKenzie. *Gamer Theory*. Cambridge, Mass.: Harvard UP, 2007. Print.
- Williams, Sam. *Free as in Freedom (2.0): Richard Stallman and the Free Software Revolution*. 2nd ed. Boston, MA: Free Software Foundation, 2010. Print.

A

- administration, 21
- Agile, 16
- Android, 20
- array, 7
- Atlassian, 16

B

- Backbox Linux, 16
- bash, 7
- best practice, 7
- Blender, 20
- break, 25

C

- C#, 20
- capture the flag, 16
- careers, 16
- certification, 16
- certified ethical hacker, 16
- cli, 7
- coding, 7
- collaboration, 16, 20
- commenting, 7
- comments, 7
- compiled languages, 7
- compiler, 7
- CompTIA, 16
- conditional, 7
- conditionals, 7
- constants, 7
- containerization, 7
- courses, 3

D

- Debian, 16
- digital forensics, 16

E

- Eclipse, 20

F

- faculty, 21

- filesystem navigation, 7
- filesystem management, 7
- flow control, 7
- for loop, 7
- forensics, 16
- functions, 7

G

- game development, 20
- game history, 20
- Games, 20
- GIMP, 20
- git, 7, 16
- Github, 16
- GNU/Linux, 16
- goals, 3
- Google, 20

H

- homework, 25
- hybrid learning, 25

I

- if/then, 7
- independent practice, 25
- information security, 16
- infosec, 16
- interactive narrative, 20
- interpreted languages, 7
- itinerary, 25

J

- JavaScript, 7, 20
- Jira, 16

K

- Kali Linux, 16

L

- learning management system, 25
- Linux, 7, 16
- LMS, 25

location, 25
loop, 7
Loops, 20
LPI, 16
lunch, 25

M

malware, 16
Mesh, 20
Microsoft, 16
Microsoft Windows, 20
Microsoft Windows 7, 7
MIT, 7
Moodle, 25
MSDE, 25

N

Narrative, 20
Network+, 16
Notepad++, 20

P

penetration testing, 16
post-secondary pathways, 16
prerequisites, 25
programming, 7
programs, 3
Python, 7

R

red teaming, 16
routine, 3
Ruby, 7

S

schedule, 3
Scratch, 7
scripting, 7
scrum, 16
ScrumMaster, 16
SDK, 20
Security+, 16
sessions, 3
shell scripting, 7
source code management, 7, 16
sprite, 7
staff, 21
STEAM Education, 3
STEAM education, 25
STEM Education, 3
STEM education, 25
summary, 3
system administration, 7

U

Ubuntu, 16
Unity Game Engine, 20

V

variables, 7
virtualization, 7

W

while loop, 7
Windows 7, 16