



STEAM Adventure 2015 @ Chelsea School

Release Draft

Chelsea School

January 22, 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
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 is the daily schedule?	24
8.8	What resources do participants need?	24
8.9	What resources are provided by the camp staff?	24
8.10	What additional resources are encouraged for participants?	24
8.11	Is homework assigned?	24
8.12	How can a participant prepare before camp starts?	24
8.13	Are there prerequisites for prospective participants?	24
8.14	Notes	24
9	Registration	25
9.1	Registration Options	25
9.2	Open Registration	25
9.3	Registration How-To	25
	Index	27

OVERVIEW

1.1 Summary

Through hands-on computing, Chelsea School's STEAM¹ Camp 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
- Set up a Raspberry Pi [credit-card sized computer]

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)

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

7.3.2 Learn More

FREQUENTLY ASKED QUESTIONS

8.1 What is 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.

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 Camp*.

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)

8.3 What is the location of the summer camp?

8.4 What are the camp hours?

8.5 What is the break and lunch plan?

8.6 How can a parent best support a participant?

8.7 What is the daily schedule?

8.8 What resources do participants need?

8.9 What resources are provided by the camp staff?

8.10 What additional resources are encouraged for participants?

8.11 Is homework assigned?

8.12 How can a participant prepare before camp starts?

8.13 Are there prerequisites for prospective participants?

8.14 Notes

REGISTRATION

9.1 Registration Options

9.2 Open Registration

9.3 Registration How-To

A

administration, 21
Agile, 16
array, 7
Atlassian, 16

B

Backbox Linux, 16
bash, 7
best practice, 7
break, 24

C

capture the flag, 16
careers, 16
certification, 16
certified ethical hacker, 16
cli, 7
coding, 7
collaboration, 16
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

F

faculty, 21
filesystem navigation, 7
filesystem management, 7
flow control, 7
for loop, 7
forensics, 16

functions, 7

G

git, 7, 16
Github, 16
GNU/Linux, 16
goals, 3

H

homework, 24
hybrid learning, 24

I

if/then, 7
independent practice, 24
information security, 16
infosec, 16
interpreted languages, 7
itinerary, 24

J

JavaScript, 7
Jira, 16

K

Kali Linux, 16

L

learning management system, 24
Linux, 7, 16
LMS, 24
location, 24
loop, 7
LPI, 16
lunch, 24

M

malware, 16
Microsoft, 16
Microsoft Windows 7, 7
MIT, 7
Moodle, 24

MSDE, 24

N

Network+, 16

P

penetration testing, 16

post-secondary pathways, 16

prerequisites, 24

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

Security+, 16

sessions, 3

shell scripting, 7

source code management, 7, 16

sprite, 7

staff, 21

STEAM Education, 3

STEAM education, 24

STEM Education, 3

STEM education, 24

summary, 3

system administration, 7

U

Ubuntu, 16

V

variables, 7

virtualization, 7

W

while loop, 7

Windows 7, 16