

Youth Competitive Programming Circle

Information, documentations, and letter of determination are included



11/18/2015

Youth Competitive Programming Circle

staff@ycpc.us

www.ycpc.us

Overview

Short facts

Name:

Youth Competitive Programming Circle

Also Known As:

YCPC

Physical Address:

316 Brower Ct, San Ramon, CA 94582

EIN:

46-5024245

Web URL:

www.ycpc.us

Contact:

Young Min Kim

staff@ycpc.us

Secretary

NTEE Category:

Youth Development

Education

Year Founded:

2014

Ruling Year:

2015

Additional information

Mission Statement

Youth Competitive Programming Circle's provides coding provisions for aspiring programming youth of all backgrounds.

Legitimacy Information

This organization is registered with the IRS.

This organization is required to file an IRS Form 990-N.

Student and Volunteer Run

Our organization is completely volunteer run, with our online courses and workshops all developed and run by student volunteers. No officers or staff members are financially compensated.

Programs

Minecraft Coding Camp

Description:

MCC is a 2 day coding workshop that uses the Minecraft ComputerCraft mod to introduce students to programming.

Website:

<https://www.ycpc.us/mcc.pdf>

A 2 day coding workshop for middle and high schoolers, YCPC's Minecraft Coding Camp program uses the ComputerCraft mod for the popular kids game 'Minecraft' to allow students new to coding to learn Lua and use robots to automate various in-game tasks ranging from strip-mining to fighting off zombies. Cutting out the intensive memorization and training characteristic of traditional pedagogical schemes, this program is designed to spark an interest in coding among high schoolers and middle schoolers.

Bridge Program

Description:

The Bridge program hosts field trips for students, enabling them to shadow developers and visit tech companies.

Website:

<https://www.ycpc.us/bridge.pdf>

Through the Bridge Program, YCPC hosts and helps other schools and clubs set up field trips to tech companies located in the Bay Area. The Bridge Program is designed to inspire more students to enter the computer science and STEM industries by organizing and facilitating trips to tech and research companies. Participants get to see that working in computer science is not just cubicle-work, but also collaboration in dynamic, team-based facilities which are famous for their great atmospheres.

YCPC Courses

Description:

Courses are a series of compact online courses in programming.

Website:

<https://www.ycpc.us/courses>

YCPC Courses are a series of online courses designed to be quick and compact without dumbing down concepts. Designed for younger audiences, the courses offer jargon-free explanations, sporadic Easter eggs and video game intermissions. These courses were created to offer a more compressed alternative to conventional courses, but without the diluted verbose rambling of online courses designated "for kids". Currently YCPC offers a Beginners Python course, with Beginners Java and Web Fundamentals courses to be released in August 2015.

Library Coding Initiative

Description:

The library coding initiative program offers materials to help transform libraries into centers of learning

Website:

<https://www.ycpc.us/lci.pdf>

Youth Competitive Programming Circle's Library Coding Initiative provides libraries with the resources to facilitate the computer science education of disadvantaged youth. Through the program, YCPC assists librarians and other interested community leaders in starting clubs, holding classes, and providing the requisite software and coding resources to allow students to jumpstart their computer science careers.

Affiliations

Microsoft

YCPC receives a free Office 365 Business suite and domain email management in a sponsorship from Microsoft.

Slack

YCPC receives a free Standard Plan package from Slack (a value of \$8 per user), an online collaboration platform, due to YCPC's status as a nonprofit organization.

IvyMax

YCPC receives a location to host its Minecraft Coding Camp and funding to purchase Intel Acer Chromebooks for the purposes of MCC from IvyMax.

Angel Academy of the Arts

YCPC receives funding from the Angel Academy of the Arts.

Luminosity & GitHub

YCPC has collaborated with Luminosity and GitHub in its Bridge Program to host educational trips for students to these companies' headquarters.

Corporation for National and Community Service

YCPC has is a certified organization for handing out Presidential Volunteer Service Awards from the Corporation for National and Community Services due to its status as a US-based nonprofit.

Guidestar

YCPC is accredited by Guidestar and is certified up to the bronze level for transparency.

YCPC Courses

YCPC features student-made course materials that may be learned with or without the support of a study group. At the moment Web Fundamentals and Python are available for use, with Beginners Java, Advanced Java, Advanced Python, and MCC Online in the works. Each course centers around video lectures that cover topics in each programming language; quizzes, study guides, and miniprojects enhance each student's understanding of the material.

Content

Every course is divided into four units that grow in sophistication and difficulty. Within each unit are three chapters, each with a video lecture; also included is one quiz, study guide, and miniproject to accompany student-learned material.

Each quiz is composed of about 30 multiple choice and free response questions that test students' conceptual understanding of the material and their practical ability to spontaneously create sections of code in response to prompts. This capability can be used to give self-studying students and instructors detailed information on the level of retention of the material.

In the event that students need to review material, study guides are available. Each summarizes and details the salient points of the topics covered in the video lectures, including references to the example code used in the video lectures. In addition, each study guide includes tips on how to improve on coding, relevant to the information covered in the video lectures.

Miniprojects are accompanying projects that extend students' knowledge to practical applications in the real world. Students write programs according to basic instructions provided on the page, incorporating recent and past topics to create functioning programs. Included is a prompt, a walkthrough that provides detailed information on the conceptual basics behind learning such topics, and answer code that students can refer to in order to assess the fidelity of their own code.

Production process

Videos are the first components made in the courses program. With such large undertakings, teams of volunteers are in charge of the various aspects of video creation. The first team writes a script, which is sent to two teams that record audio and video are recorded separately but simultaneously. These are then sent to the editors who splice the product together and edit it for public use. Everything is then uploaded to the PHPMyAdmin database and published on our website.

Study guides, miniprojects, and quizzes are produced by a specialized team to accompany the material covered in the video lectures.



MCC DOCUMENTATION

An overview of the Youth Competitive Programming Circle's Minecraft Coding Camp,
including curriculum and project details

Youth Competitive Programming Circle
staff@ycpc.us



Minecraft Coding Camp: A Virtual World Application

YCPC's Minecraft Coding Camp Program uses ComputerCraft mod for Minecraft to allow students new to coding to learn Lua and use robots to automate various in-game tasks.

Cutting out the intensive memorization and training characteristic of traditional pedagogical schemes, this program is designed to spark an interest in coding among high schoolers and advanced middle schoolers.

Minecraft Coding Camp

The Minecraft Coding Camp teaches kids to code using a game modification for Minecraft (referred to as mod for short) by the name of "ComputerCraft". This mod enables students to program robot automation in-game with the Lua programming language, allowing beginners to receive immediate feedback on their programming in the gaming context they are familiar with.

In ComputerCraft, students play on a hand-crafted adventure-style world. They are guided through an in-game tutorial of the basics of coding in Lua: print statements, if-else statements, loops and given documentation on how to use such commands. Then the students are free to choose from a structured curriculum of lesson-projects, each of which is a mini-adventure that increments in difficulty and teaches the student to master built-in robot functions such as `.moveForward`. This will culminate in the ability to automate their "turtle" robot to complete any task ranging from farming to fighting off zombies.

With a 2:1 student to instructor ratio, no students are left behind and are provided significant assistance with the creation of their own programs.



Minecraft Coding Camp – An Overview

Advantages over Traditional Coding

One of the primary hurdles for beginners to coding is realizing the value of software development. For example, for the Lua programming language, it takes the complex installation of the Lua development kit, runtime environment, development interface, and requisite libraries to even begin coding. Then it takes six lines of syntax-sensitive code and compilation to make the computer write the simple "Hello World".

ComputerCraft, on the other hand, uses virtual robots and their corresponding built-in functions to allow students to program a zombie-fighting machine in those same six lines of code. This makes coding a more meaningful, humble, and entertaining endeavor.

Program Structure

A typical camp consists of two hour sessions split across 2 days. Camps are typically held after school.

Tutorial Students are guided through an in-game tutorial, introducing basic coding concepts and giving a brief overview of the heavily-used coding logic. Students are given a reference book on Lua syntax that may be used later on.

Projects Each of the 4 projects is a mini-adventure where students have to code robots to overcome specific obstacles, learning both robot-specific functions and implementation of logic in "real"-life situations.

Freeplay After each project, students are teleported into a special arena with varying environments where they apply the functions and syntax learned throughout the project's "mini-adventure" to write their own code.

In a typical day, a session begins 15-30 minutes after the end of school and lasts for 90 minutes. The first 30 minutes of the program serve as an orientation where students sign in and are introduced to the Minecraft and ComputerCraft interface. The next 45 minutes are dedicated to students completing the in-game tutorial and trying out the basic syntax and coding logic on the ComputerCraft interface. During the remaining 30 minutes of the first session and the entire second session students can work on any of the four projects. Because a typical project takes around half an hour to complete, students should be able to experience and experiment with all four of the provided projects.



Curriculum

ComputerCraft basics

- ✓ Operating a system console
- ✓ Creating, saving, and running programs
- ✓ Printing text to console
- ✓ ComputerCraft “Turtles” and their special functions

Variables

- ✓ What is a variable
- ✓ How to use a variable
- ✓ Comparing variable values
- ✓ Incrementing variables to control loops

If Statements

- ✓ “if” statements
- ✓ Using functions as conditions
- ✓ Complex/nested “if” statements

Loop Statements

- ✓ While loops
- ✓ Infinite loops
- ✓ Nested loops
- ✓ Using loops to iterate through a 2D grid

Functions

- ✓ Defining and calling local functions
- ✓ Conditional functions
- ✓ Fail-safe movement functions

Prerequisites

On-site Needs

The Minecraft software requires somewhat powerful computers. Also, because our program requires a modded version of Minecraft (to install the ComputerCraft mod), our organization will need access to either the Roaming folder (Windows) or the Library/ApplicationSupport folder (Mac)

Student Experience

Although our program introduces the basic concepts, some students may find the coding logic confusing. Those who need extra assistance should first reference and familiarize themselves with the conditional logic used in coding (such as if-else statements). There are countless resources for this online, including YCPC’s personalized online courses at <https://www.ycpc.us/courses>.

Continued Education

We highly suggest that students continue their coding education after MCC. Along with online resources and our online courses, our organization also offers the Library Coding Initiative and a Chapter program to create a local community of instructors, peers, and teachers that will assist students.



Tutorial

The tutorial for MCC is divided into four rooms that teach the basics of Minecraft to students that are unfamiliar with the game. This tutorial does not require teacher intervention, and has text that automatically pops up on the computer screen to guide the student. After each room, the student's character gets automatically transported to the next room.

Room One – Basic Controls

This room teaches the fundamentals of playing Minecraft such as movement and inventory access. Conveyor belts are laid around the room to ensure that no student gets lost or has trouble finishing the room.

Room Two – Computers

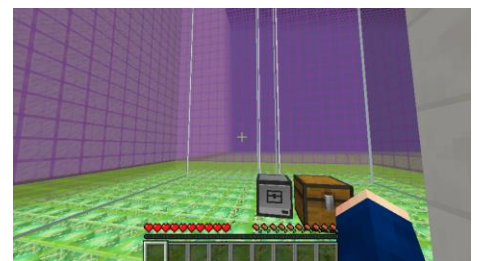
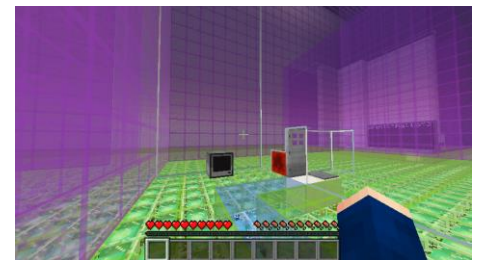
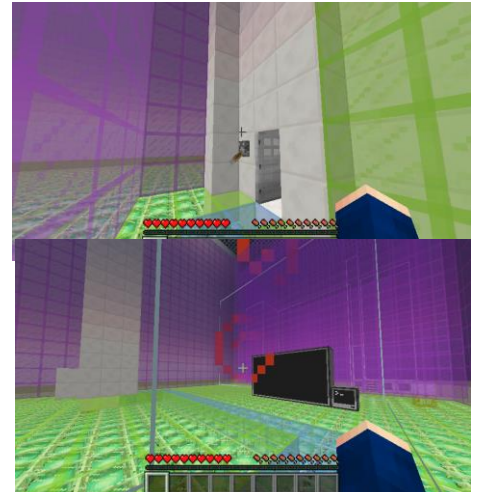
To run programs in Minecraft one has to utilize computers, which serve as the compilers for coding in Minecraft. This room allows a student to accustom oneself with how to run programs using a Minecraft computer. The 'run' keyword is taught in this room.

Room Three – Creating Programs

Students in this room are introduced to creating, saving, and editing their own programs using the Minecraft computer. The 'print' keyword is taught in this room.

Room Four – Commands and Turtles

Students in this room delve further into understanding basic commands in Minecraft. Specific syntax commands that may be unfamiliar to students learning programming for the first time are dissected to ensure the student fully understands how the command syntax works. Students are also introduced to turtles, which are the Minecraft robots that will be the subjects of all the coding.



Project I - Mining

After the tutorial, students will be able to work on one of four adventure-map styled projects.

In the first lesson students are guided through programming an automated mining "turtle" that will mine for resources and move to find new ores. Using the basic turtle functions `.dig`, `.turn`, `.refuel` and `.move`, students will learn the basic information needed to develop more complex automatons. Starting out with mining through basic obstacles, students eventually have to develop software that automatically digs through a 2D grid to create a quarry. More specific details are listed below.

The mining project starts off with accessing a program called 'mine', which is located in a mining turtle. After understanding the digging code, the student is asked to run that program and see the turtle mine one stone block.

Then the student proceeds to the next part, where a program is already precoded to mine two stone blocks stacked on another.

The first task the student must complete independently comes in the third room. The student has to code a program that digs a 1x1x2 hole through a stone wall in order to progress. Using the `refuel`, `forward`, `dig` and `up` commands utilized in the previous 'mine' programs, students must figure out how to proceed.

The tunnel that the student made will lead to another requiring a student to get across a pool of lava using a turtle's `place` function. Many new concepts are required for this task, including assigning variables, checking variables (`==` sign), `detect` functions, `if then` statements, and `place` functions.

After getting past the treacherous lava, the student must dig through another stone wall before getting to the next challenge, where the student must dig a redstone block located on top of lava using `turtle.forward()` and `turtle.dig()`.

Finally, the student has to try to get as many emeralds as possible using only a mining turtle in the 'arena' below. This task encompasses all the commands learned thus far, like `refuel`, `forward`, `dig`, `digUp`, `digDown`, etc.



Project 2 - lumberjack

In this project, students automate a robot to cut trees of varying heights and replant saplings. The lumberjack automation uses functions from the mining tutorial and also introduces slightly more sophisticated utilities such as .detect and .place.

Loops and arithmetic operators are introduced at the start of this project, where students get familiar with code for cutting down trees of any height using logging turtles.

Then the student is asked to obtain a redstone torch located at the top of the tree. He or she will have to use while loops and dig, digUp, and place functions.



Project 3 - Melee

Students synthesize previously learned concepts and ideas to create automated guards that fight off zombies and hunt down monsters. Students learn to automate melee turtles to complete various tasks such as defending the player or moving in a snake pattern to kill all mobs in a 2D grid.

Students start off by learning the code for killing zombies in a confined 2x2 space. The code needed involves while loops and attack functions. Local functions are also introduced in order to reduce code so that students don't repeat the same code over and over. Local functions and while loops are used to move the turtle around in a circle to kill all the zombies.

To proceed, the student must create a program that kills all the zombies within a 2x5 grid. Students will have to use some ingenuity as well as the previous code given to complete this task. After killing the zombies, the student can then obtain the redstone block in the chest that the zombies were guarding, and proceed to the next challenge by placing the redstone block in the designated spot, which opens the door to the next area.

The last challenge in this project is to kill 10 zombies that spawn inside an arena. Students will not be able to attack them head-on, so they will use the two turtles at their disposal and create a program so that they will go around the arena and kill the zombies. Student knowledge will be put to the test as students need to figure out how to utilize while loops and local functions to program turtles that will kill the zombies sufficiently. There are many different ways and strategies for completing this task.



Project 4 - farming

Here students automate a turtle to suck seeds from a chest, till dirt, plant seeds, harvest crops, and place the yield into a chest. To do so, students utilize advanced features such as .suck, .drop and .sleep. This farming project is by far the hardest project, and is designed for the most motivated of students.

The first part of the project demonstrates a basic tilling system using the farming turtle. The program shown uses variables, if/then statements, modulus, and moving functions. The turtle snakes through the plot of land, tilling the land as it goes by. The program itself keeps track of the current row and current column the turtle is in, and implements the necessary movements to make sure every tile of land is tilled. There are four 'demonstration' turtles, each of which introduce more and more features, like seed planting, harvesting and storage of wheat, and the bonemealing of wheat. However the basic movement code remains the same for all four turtles.

Then the student comes to a small plot of land, where he or she can put all their knowledge about farming in Minecraft to use. The student will need to utilize the movement code demonstrated in the four turtles' code beforehand, as well as implement various other features like seed planting, and wheat harvesting, storage and bonemealing. This task is very open-ended and will be very challenging.

If the student has completed all the projects in the specified order, after this project, the student has completed the camp!



Bring Your Own Device:

At some sessions, computers may be provided; however, if YCPC computers are unavailable, we do ask that students bring their own devices. In such a situation we ask that they bring a computer capable of running relatively demanding software. Also, because Minecraft burns through battery rapidly, we highly recommend the presence of a compatible charger. Wi-fi will be provided at events unless otherwise noted.

Set Up Process

If students use their own computers for the program, some software will need to be installed.

Required Components:

- ✚ Minecraft Launcher:
 - Minecraft.exe for Windows
 - Minecraft.dmg for Linux
- ✚ Minecraft Forge Version 10.13 (To run Minecraft Mods)
- ✚ Computer Craft Mod
- ✚ Custom MCC World
- ✚ Minecraft Account

Installation Process:

The standard process for installing the required components

- i. Place our custom Minecraft folder with Forge, the Computer Craft mod, and our custom world preinstalled. In Mac, replace the Minecraft folder in Library/Application_Support. In Windows, replace the .minecraft folder in Roaming (can be found by searching %appdata%)
- ii. Run the Minecraft launcher for your specific operating system, which can be found at <https://minecraft.net/download> or provided on-site
- iii. Students may use their own Minecraft account or an organization provided account.





Library Coding Initiative

Youth Competitive Programming Circle
staff@ycpc.us



The Library Coding Initiative

Youth Competitive Programming Circle's Library Coding Initiative provides libraries, districts, and cities with the resources to educate disadvantaged youth in computer science.

Why libraries?

In impoverished areas, a significant number of children have no access to computers or the Internet at home: for these individuals, the library serves as the primary connection to the internet. Consequently, the library is a crucial piece of infrastructure that disseminates the vast plethora of knowledge available on the Web to students. LCI piggybacks on this function to provide students with technology-related education. In doing so, YCPC will assist librarians and other interested community leaders in starting clubs, holding classes, and providing the requisite software and coding resources to allow students to jumpstart their computer science careers.

Ways to Get Involved

1. Contact your local library

Reach out to your nearby library or librarian and ask them to get involved. If you are a librarian or a community leader, please send an email to staff@ycpc.us. If you have any events or workshops we would love to assist with our enthusiastic staff members and programming resources.

2. Get your school involved

Consider starting a chapter – it's a fun and easy way to jump into the world of coding. If you're interested on getting your peers into the Library Coding Initiative, feel free to read our helpful guide to running a chapter at www.ycpc.us/documentation and reach out to us at staff@ycpc.us.



Youth Competitive Programming Circle's official chapter documentation is also a useful reference for libraries:

<https://www.ycpc.us/documentation>

For more information regarding our Minecraft Coding Camp program, please visit:

<https://www.ycpc.us/mcc.pdf>

School Districts

Through the Library Coding Initiative program, Youth Competitive Programming Circle partners with school districts to improve student awareness of available educational resources both online and in their local library. With the help of teachers and student leaders, learners will be exposed to coding workshops and other educational experiences. Students who seek to further their education will know where and how they can get the help they need to pursue an education in STEM.

Libraries

Youth Competitive Programming Circle provides a reference packet full of free online course providers for students to learn coding by and also instructions on how to install developer interfaces on library computers. Also, YCPC provides a useful chapter documentation tool full of easy-to-hold events for use by clubs and libraries alike, whose contents range from study groups to Minecraft Coding workshops. With a detailed process, librarians even without a background in computer science can hold these events and help attract and expose more students to the libraries educational resources.



Coding Tutorial Resources

- **YCPC Courses** (<https://www.ycpc.us/courses>)

Tailored to kids in middle and high school, YCPC courses are specifically designed to be user-friendly and instructive, with hands-on mini-projects, periodic quizzes and useful video lectures.

- Difficulty: Medium
- Cost: None
- Languages available: Beginners Python, Java (at a later date), Web Fundamentals (at a later date)
- Target age group: Adolescents attending middle or high school

- **Udacity** (<https://www.udacity.com>)

Sponsored by technological institutions like Facebook, Google and AT&T, the courses at Udacity satisfy every programming aspiration by covering a wide variety of programming languages.

- Difficulty: Beginner-Advanced
- Cost: None (\$200/month for Nanodegree programs)
- Languages available: Web, Android, iOS, Java, JavaScript, Python, among others
- Target age group: High school and college students

- **Codecademy** (<http://www.codecademy.com>)

Codecademy is perfect for novices who want interactive and easy-to-follow tutorials.

- Difficulty: Beginner-Medium
- Cost: None
- Languages available: HTML, CSS, JavaScript, jQuery, PHP, Python, Ruby
- Target age group: All ages

- **Treehouse Club** (<https://teamtreehouse.com/>)

The comprehensive courses provided by the Treehouse Club are excellent for serious students.

- Difficulty: Medium-Advanced
- Cost: \$25/month (Basic Plan); \$49/month (Pro Plan)
- Languages available: HTML, CSS, Ruby, Android, WordPress, Python, Java, PHP, iOS
- Target age group: College students and older



How to Install Various Coding Languages

In order to convert user-inputted code to code that the computer can understand, many people use compilers to create executable programs. In addition to compilers, some languages also need separate executable applications in order to start programming. Here are some tutorials for some programming language compilers:

Python

1. Go to <https://www.python.org/downloads/>
2. Click on "Download Python 3.4.3". Execute the downloaded application.
3. A command prompt box should pop up. Have fun coding in Python!

Java

For setting up Java, you will need the Java Development Kit (JDK) and a compiler called Eclipse IDE.

1. Go to <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
2. Then, click the "download" button located under the word "JDK".
3. Under "Java SE Development Kit 8u45", accept the license agreement, and then select the version of JDK that matches your operating system.
4. Download the installer and run it to install JDK.
5. To install Eclipse IDE, go to <http://www.eclipse.org/downloads/packages/eclipse-ide-java-developers/marsr>
6. On the right, download the appropriate file that matches your operating system.
7. Extract the file with a tool like WinRAR to a familiar location.
8. Once the file has been extracted, go into the folder and make a shortcut of the Eclipse application and put it in your desktop for convenience.
9. Open Eclipse IDE. Have fun coding in Java!

Xcode (Mac OS only)

1. Go to <https://developer.apple.com/xcode/downloads/>
2. Download "Xcode 7 beta 3"
3. You will be prompted for your Apple ID and password
 - a. If you have an Apple ID: input username and password.

- b. If you do not have an Apple ID: create an Apple ID with the required information and proceed.
4. You will be at the Apple Developer website. Head on over to <https://developer.apple.com/xcode/downloads/>
5. Click the link to download "Xcode 7 beta 3"
6. The download is very large, so it may take a long time.
7. Open the .dmg file once it has finished downloading.
8. A dialog box will pop up for you to install Xcode in your Applications folder. Do so by dragging the Xcode icon to the Applications folder.
9. Once Xcode has finished copying to Applications, open Xcode directly from Applications.
10. Be sure to acquire admin password in order to consent to the Xcode license agreement.
11. Xcode is now open. Have fun coding in Xcode!

VB.NET Express (Windows only)

1. Head to <https://www.visualstudio.com/downloads/download-visual-studio-vs#d-express-windows-desktop>
2. Click "Install Now" under "Microsoft Visual Studio Express 2013 with Update 4 for Windows Desktop – English".
3. After providing the needed details click on the "Your Selection" link.
4. Once wdexpress_full.exe has finished downloading, launch the installer.
5. Proceed through the installer's instructions. When the installer is done installing click "LAUNCH"
6. Click "Not now, maybe later" when you are prompted to sign into Visual Studio.
7. VB.NET installation has now finished. Have fun coding in VB.NET!

WAMP and AMPPS (for HTML, CSS and JS)

For Windows – WAMP installation:

1. Head to <http://www.wampserver.com/en/>
2. Scroll to the bottom of the page. There should be a "Downloads" section. Select your current operating system.
3. Once the installer has finished downloading, open the installer and follow the instructions.
4. To start your first WAMP server, open the WAMP application. Once opened, a WAMP icon will appear in your taskbar. If it's green then all services are working. If it's yellow then some services are working. And if it's red then no services are working.

NOTE: usually you need to give the server a minute to transition from red to green.



5. Once the icon is green, your WAMP server is working. If not green after a few minutes, right-click the icon and select "Restart All Services".
6. Have fun coding in HTML, CSS and JavaScript with your new WAMP server!

FOR MAC and LINUX– AMPPS Installation

1. Head to <http://www.ampps.com/downloads>
2. Scroll to the bottom of the page. Select your current operating system.
3. Run the installer and follow the instructions.
4. AMPPS is now done installing. Have fun coding in HTML, CSS and JavaScript with AMPPS!

Ways Libraries Can Increase Computer Access for Programming Purposes

1. Allow all kids to use computers for programming

All kids that come to the library will not have to go through the process of applying for a library card to learn how to code. This change will make the students' coding experience much more accessible.

2. Preinstall development kits and development environments/compiler for library computers

The process of installing development kits and compilers can be convoluted at times. One way to make the experience of learning code more efficient is to preinstall all the development kits and environments/compiler on every library computer. Using the instructions detailed in "How to install various coding languages", librarians can install the requisite programming tools ahead of time.

3. Provide helpful coding references as browser bookmarks

Default bookmarks that serve as references for coding will be helpful for learning unfamiliar syntax one of the major stumbling points in learning programming.

4. Provide a convenient method for kids to save their code via USB

Students will be able to conveniently export all the code they have worked on to a USB to work on at home or the next day at a workstation, as library computers often cannot save files.



Managing Study Groups

YCPC Study Groups are a community based schema of studying YCPC courses. Below is a rough outline of what a group can do.

Study groups meet weekly for discussion and projects; to prepare for meetings, members view any video lectures, quizzes, study guides and other studying materials at home utilizing the online course material.

At library meetings, students collaborate on the given project/challenge. Students collaborate and add features to the program as a group: a difficult but necessary skill in larger teams of developers.

A librarian is also needed to oversee the study group. They lead the discussions, ensure member attendance, keep the environment positive, and make sure the study group isn't going off track.





MS BRIDGE PROGRAM DOCUMENTATION

An overview of the Youth Competitive Programming Circle's Middle School Bridge Program. Highlights logistics including location and dates in addition to safety measures to provide the best experience.

Youth Competitive
Programming Circle
staff@ycpc.us



YCPC Bridge Program

Connecting Students to the World of Technology

Culture of the Computer Science Industry

One of the hallmarks of technology companies these days are the perks and benefits they offer to their employees.

Since technology workers are in high demand and generally short supply, companies will often bend over backwards to entice workers to join them, by offering high salaries, fun offices and good food.

The Bridge Program is designed to inspire more students to enter the computer science and STEM industries by organizing and facilitating trips to tech and research companies. From setting up guided tours with software startups to organizing lunch reservations, this program assists schools in creating easy and rewarding field trip experiences with YCPC handling much of the logistics for students in order to demonstrate the practical benefits of working in a rapidly growing sector.

Students will be able to see that working in computer science is not mere cubicle-work, but collaboration in dynamic, team-based offices famous for their great work atmospheres. Students will not only be exposed to unique workplaces, but also interact and follow real developers as they go about their daily work.



YCPC Bridge Program

A First Hand Look

One of the largest misconceptions regarding computer science is that it's merely the art of smashing away at keyboards in cubicles. This program is geared towards showing students the workplace of a company at the cutting edge: bright offices filled with food, unorthodox decoration, and a productive, happy workforce. It's this idea that the program intends to show to students, and hopefully entices more computer science students with.

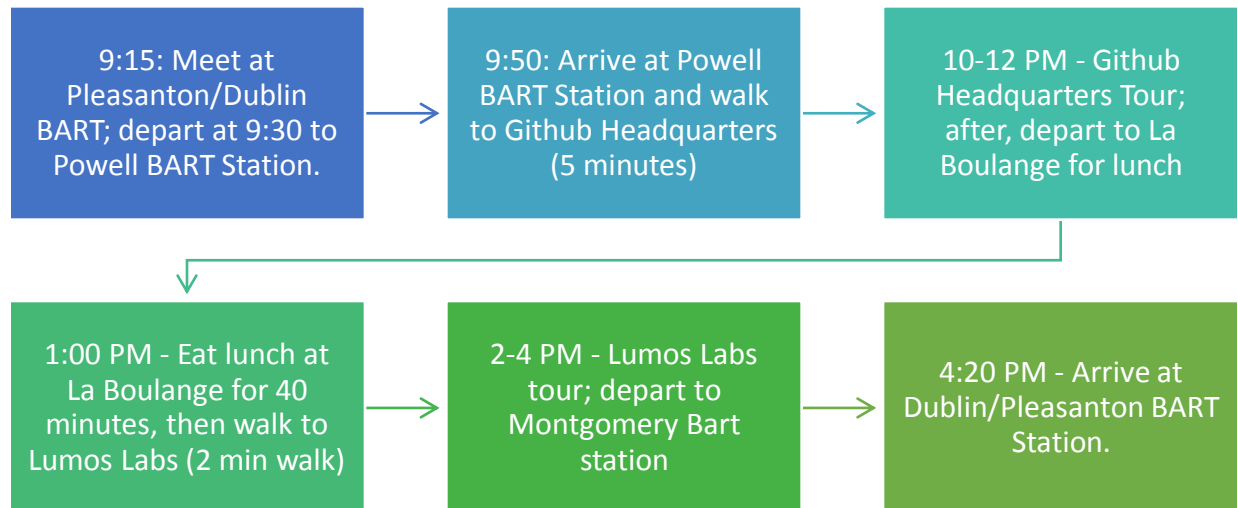
A Typical Bridge Trip

Most trips take place on weekdays around 9 o'clock as many companies are closed over the weekend. Beginning with a guided tour at the first company, students will be able to shadow developers and observe firsthand the workday at a tech company. Around noon, students leave the company and head to a nearby restaurant for lunch. YCPC does not organize transportation, but is able to reserve restaurants within walking distance. Students will then attend the next company and repeat the process as before. The tour will end around 5 PM, and students will get home around 5:30 PM.



MS Bridge Program

Sample Schedule from Previous Excursion



Prerequisites

There are no prerequisites for students to participate on a program: all skill levels and backgrounds are welcome. If there are any students with unusual circumstances that intend to attend, we request that we are informed prior in order to accommodate their needs.

We do request that the school provide transportation to and from the offices. Due to liability issues, the organization is unable to organize such services for the students. Although we do try to make sure that companies and restaurants are within walking distance of each other, commute between the trip location and the school may vary to up to half an hour.

Although student volunteers will be present to assist with the trip, they will not officially be part of the trip and will not be able to lend students money or be obligated to be chaperones.

