Saint Ann's School

HIGH SCHOOL COURSE CATALOG 2014–2015

PRINTMAKING: POSTERS

(Lee)

This course is devoted to poster design and production. A historical survey of poster designs includes: Japanese nineteenth century playbills, Polish circus posters, Mexican revolutionary leaflets, rock posters of the sixties, and more. This course also works with the Theater Department to produce the posters for school productions throughout the year. Various printing techniques are explored.

CERAMIC SCULPTURE

(Bellfatto)

Not a pottery course. We explore basic clay building techniques such as coil, slab and pinchpot to generate functional and non-functional sculpture. Various surface treatments are investigated: stain, paint, and glazes. Students develop a body of work reflecting an eclectic variety of sources and themes: personal, historical, geometric and organic form, human and animal figure, narrative relief, and architecture.

SCULPTURE

(The Department)

A broad course for both beginning and advanced students. The class investigates the three-dimensional form as a medium for self-expression. This exploration uses a variety of sculptural techniques and materials, such as modeling from life in clay and wax, plaster casting and carving, and wood constructions.

COMPUTER

We humans are surrounded by the stuff that used to be science fiction—portable computers, artificial intelligence, 3D printing, electronic games, online journals, instant reference books, genetic sequencing, video and still cameras everywhere, nanotechnology, increasingly massive datasets—with more innovations coming backed by data processing, design, number crunching, and computer science. Our courses show students how to be more than just consumers or users: they will be independent creators on computers, able to control and help shape the tools of today and tomorrow. Using software that runs similarly on Mac, Windows, Unix/Linux, and tablet computers, our courses teach a range of topics including programming, graphics, circuitry, web, spreadsheet analysis, logic, and other skills that are useful for doing everything from analysis to artwork. Classes are full year and meet twice a week (unless otherwise noted) using desktop and tablet computers. Visit tinyurl.com/sacc2014 for more information about any of these classes.

3D PRINTING

(Arum)

3D printers are personal fabrication tools that are a part of an evolving modern world of technology that allows students to become producers, inventors and artists. Students will create, design, invent and prototype while efficiently and inexpensively taking their digital designs into the real world. Students will be able to easily understand the strengths and limitations of their work and will be encouraged to modify their designs, thereby participating in an iterative engineering design process. Students will learn various 3D modeling techniques and explore several 3D modeling software tools and packages.

ALGORITHMS FOR GENETIC SEQUENCING

(Roam)

For experienced programmers, this class introduces programs that analyze genetic sequences. There are numerous exercises in pattern-matching and string comparisons, calculating family trees based on DNA sequences while taking into account the basic operations of mutation, insertion, deletion, and transposition. Though we mostly use simplified models of DNA (without worrying about protein folding), this topic gives us a chance to study "design patterns," data-structures and algorithms for large data sets, and basic molecular models. **Prerequisite:** two programming courses.

ANIMATION ON COMPUTERS

(The Department)

In this class, learn about computer-aided methods of animation. Use Flash to complete frame by frame animations, including the traditional walk-cycle project. Use AfterEffects for more advanced techniques such as pinning digital puppets and working within a 3D space. Create special effects such as lightning and explosions. Additional projects may include stop motion, green screen projects, 3D movie titles (like the iconic Star Wars titles), music videos, and a final animated movie using techniques of the student's choosing. No prior experience is required, though attention to detail and perseverance are a must!

COMPUTING I

(The Department)

The basic information necessary to use computers for schoolwork, learned through creative independent projects. We emphasize the use of computers as aids to writing and research (typing, word processing, database, spreadsheet), programming, desktop publishing, web page design, animation, and telecommunication.

DYNAMIC WEB DEVELOPMENT

(Hoke)

Students who complete this course will be able to create customized blogs powered by Tumblr or Wordpress, eye-catching portfolios, or simply fun novelty sites (such as snowdaycalculator.com or the recently popular drakeweather.com). In addition to developing a strong foundation with HTML5 and CSS3 (the basic scripting languages in web development), students will also learn how to use JQuery for interactivity and animations, templates and frameworks such as Bootstrap and Foundation for quick design and development, and PHP and MySQL for powerful backend functionality. We will discuss responsive design, so your projects will look great on any screen large or small. Plugins and web APIs will also be covered so you can include Twitter, Instagram, Flickr, Vimeo, YouTube and other feeds on your web pages. A willingness to read documentation and research solutions is the reality of a web developer, and students should be prepared to "google" regularly.

ELECTRONIC MUSIC MACHINES

(Hoke)

Are you a fan of electronic dance music? Do you drool over professional launchpads and drum machines in music stores? Have you ever wanted to literally march to the beat of your own drum? In this class, we will investigate electronic music creation through the construction of sequencers, drum machines and MIDI controllers. Technologies we will cover include basic Arduino programming, circuit design, interface design, and an introduction to MIDI messages and music composition.

EXHIBITION SPACE

(Arum)

In this course students will collaborate on a series of temporary, site-specific interactive installations in the Ettl Lobby. These installations will respond to the viewers' physical movements and engage them as participants in generating imagery. In class students will explore programming with Processing, OpenFrameworks, and OpenCV, and will be encouraged to experiment with capturing images using cameras, the Kinect and a variety of other sensors. We will also read about and look at examples of site-specific art and installations to provide a grounding for our conceptualizations. **Prerequisite**: no prior programming experience, but a willingness to work hard is required.

GAME PROGRAMMING

(The Department)

Designing games presents unique challenges distinct from the design issues of other interactive media. In addition to the user interface, one must consider story, culture, modeling, and implementation. This course will explore developing usable and engaging games, human computer interaction, thematic structures, graphic design, sound effects, and game aesthetics. The course will take into account the history of non-digital and digital games, role-playing, puzzles, interactive fiction, and 3D modeling. Students will plan and create games both individually and collaboratively using a variety of languages and tools such as Unity3D, Livecode, Javascript, Java, and Python. We'll explore the creative possibilities of game design and experience the beauty and logic of programming. **Prerequisite:** some programming experience or permission of the instructor.

GENERATIVE ART

(Arum)

Codes are instructions, information, rules, and routines that govern our lives and also our technologies. As long as there have been codes, there have been artists to break them, play with them, and reinvent them. This course will explore how to use code creatively to generate artworks and to understand how artists make codes, use codes and break codes in the process of creating art. In addition to learning how to program with processing, we will also look at nstructional procedures, conditional design, self-imposed constraints, generative art, and parametric design. No prior programming experience is necesary but a willingness to work hard is required.

GRAPHICS PROGRAMMING

(Roam)

In this course, we write programs that create 3D computer graphics (houses, robots, landscapes). Once we complete a brief introduction to matrix multiplication, we can start shading, rotating, and animating objects that we have designed. Our programs read and process text files that contain descriptions of 3D graphic objects and display the resulting 3D objects from arbitrary viewpoints. For advanced students, projects include the construction of race car and airplane games with first person and chase plane viewpoints. **Prerequisite:** Programming 1 or permission of the instructor.

IPHONE PROGRAMMING

(The Department)

Learn how to program with Objective-C, Interface Builder, and XCode on the iPhone and iTouch's unix-based operating system. Understand the way the iPhone applications work and how to build them. Actively and creatively explore this new field of little computers using the iPhone as the main research platform. No iPhone required. **Prerequisite:** some programming experience.

PHYSICAL COMPUTING 1

(The Department)

Learn how to interact physically with a computer without using the mouse, keyboard and monitor interface. Move beyond the idea that a computer is a box or a system of information retrieval and processing. Using a microcontroller, a single-chip computer that can fit in your hand, write and execute interactive computer programs that convert movement into digital information. Work with components such as resistors, capacitors, diodes and transistors as well as integrated components. Through lab exercises and longer creative assignments learn how to program, prototype and use components effectively. Control motors and interpret sensors, as well as explore advanced concepts in interface, motion and display. **Prerequisite:** some programming experience or permission of the instructor.

PHYSICAL COMPUTING WORKSHOP

(Arum)

Creating interactive work relies on building a relationship between the object and the viewer. By gathering information in the form of input, processing that into meaningful data, and outputting that contextually, new forms of engagement and interaction with an audience can be established. This class will focus on the input side of physical computing by researching various sensors and sensing methods and developing example methods for their use. By the end of the year, the class will assemble a library of sensor applications for interactive applications. Topics will include: different types of sensors, time and events, amplification and filtering of sensor signals, sensor networks, data processing, gesture recognition, serial and wireless communication, I2C, and related topics that aid in making sensor systems effective. By using Arduino, Processing and C++, students will also further develop their programming skills. There will be short one-week exercises that students complete to demonstrate the techniques discussed in class. In addition, students will be responsible for a sensor research project in which they explain the operating principles of a given sensor and present a working example of the sensor in use. **Prerequisite:** Physical Computing or permission of the instructor.

PROGRAMMING 1

(The Department)

Explore the science and art of computer programming. For students who want to create and modify their own computer software, this course uses the high-level programming languages Java (an internet-savvy version of C++) and Livecode (a multimedia descendent of Hypercard) to introduce the basics of computer control. We use loops, variables, procedures, input, output, and branching decisions (with Boolean logic) to control graphics, sounds, and information.

PROGRAMMING 1 (INTENSIVE)

(The Department) (4x per week)

Explore the science and art of computer programming. Learn important problem-solving and design strategies like modularization and iterative design which can apply to both programming and non-programming environments. This intensive, four periods per week class is for students who want to master fundamental programming concepts which include loops, variables, procedures, input, output, conditionals and data structures. Assignments will allow students to control graphics, sounds, and data while also encouraging them to think creatively, reason systematically, and work collaboratively.

PROGRAMMING 2

(The Department)

A continuation of Programming I, for students who are becoming more confident in their ability to combine data types and complex computer routines. We use Java (an internet-savvy version of C++) to look more deeply at object-oriented programming: class definitions, inheritance, methods, fields, arrays, and collections. Large projects include writing an interactive, animated project with control windows and graphics. **Prerequisite:** Programming 1 or permission of the department chair.

SOFT CIRCUITS: WEARABLE, SOFT AND EXPERIMENTAL CIRCUITS

(Arum)

Standard electronic components can be hard, brittle or difficult to work with. Embedding them in soft environments, like clothing or toys, can be awkward. Building circuits using paper, fabrics, thread and paint opens up new possibilities for soft, curly, organic, visible and attractive electronics. This class will explore materials, components and construction techniques for successfully integrating soft materials with standard electrical supplies. The results will be light, thin, flexible, durable, aesthetic, and even expressive circuits. We will also cover techniques for integrating the Arduino Lilypad microcontroller and XBee radio communication to create interactive and social objects. Students will develop wearable devices and accessories as a means of self-expression and communication. Explore the relationship between the body, fashion, technology and social interaction. Experiment with materials and objects, and develop concepts to refine, construct and test. Make t-shirts that interact with each other, objects that light up in the dark, toys that talk to each other or artwork that responds to a user's movements. The possibilities are almost endless.

TOYS: USE, MODIFY, CREATE

(Arum)

Hack existing toys, take advantage of microcontrollers and research, design, prototype and test your ideas for toys using both digital and non-digital materials. This is a hands-on class that will teach you about sensors and actuators, microcontrollers, 3D modeling, fabricating and how the toys you make may not just be for play, but could be an implementation of therapeutic and assistive devices.

VOXEL ENGINES

(Poindexter)

Imagine an entire world small enough to carry in your pocket. Are you with me? If you can imagine it, you are almost there. "Voxels" make this possible; they are not magic; they are, in fact, totally logical. Imagine the world is made of little chunks. These chunks are volumetric pixels or voxels. Voxel engines are used to represent 3D data in scientific applications and to create procedurally generated terrain in the game Minecraft. In this class, we will have one project for the whole year: to create a voxel engine that procedurally generates terrain from the ground up in Java. We will start by learning to make a single block out of a set of points in 3D space, to package these points as a mesh and to get them to show up on screen. From there, we will add the ability to change the appearance of blocks, build randomly generated surfaces out of clumps of blocks and add caves and biomes. As part of our efforts, we will encounter concepts and strategies that are useful in programming in general. Having a basic understanding of object-oriented programming is a prerequisite for taking the class. **Prerequisite:** Programming 1 or permission of the instructor.