**Computer Science I**

*** operator** – See **operator**.

**+ operator** – See **operator**.

**- operator** – See **operator**.

**/ operator** – See **operator**.

**Active window** – The window that is affected when you press keys on the keyboard. When you switch windows between software programs, you change which window is the active window.

**Address** – The location of a file or folder on a computer. Also, the location of a web page on the Internet.

**Agile software development** – A way of organizing programmers into small teams who focus on small parts of a software program.

**Alias** – See **shortcut**.

**Alphanumeric character** – A character that's either a letter or a number.

**Alpha release** – The phase of a software project in which the software program is usable enough to test.

**Angle** – The measurement in degrees of the distance between two lines at the point where the lines connect.

**append() function** – A method that adds a new element to the end of a list.

**Appending** – Adding on to something.

**Application** – A program that focuses on doing one type of thing, such as word processing, email, or web browsing.

**Argument** – A letter, number, word, or combination of those that a command needs in order to work. Each command will only accept specific types of information as arguments. A function needs arguments in order to work correctly.

**Assignment** – Giving a value to a variable.

**Assignment operator** – An operator that changes the value assigned to a variable.

**Attribute** – A value assigned to an object.

**Autocomplete** – A feature that is meant to help you find and complete the code you are typing. Autocomplete is also called code completion.

**Automated testing** – A way of using software programs to test another software program.

**Backslash** – The \ character.



© Copyright 2010 Giant Campus

**backward()** – In the turtle program, the backward() command moves the turtle backward from where it's currently facing.

**Beta release** – The phase of testing which focuses more on the usability of the software and in which the software is distributed to a larger group of people familiar with the software project.

**Beta tester** – A customer who gets a free copy of beta software for testing.

**Bidirectional** – Moving in two different directions. See also **unidirectional**.

**Block of code** – Lines of code that work together to do one thing. A block of code is sometimes called a code block.

**Boolean data type** – A data type that can have the values True or False.

**Bug** – An error in a program's code. See also **debugging**.

**Button** – A clickable graphical object that makes something happen when the user clicks it, usually with the left mouse button.

**bye()** – In the turtle program, the bye() command will close the turtle graphics window as soon as the ENTER key is pressed.

**C** – A popular programming language that was developed in the early 1970s by Dennis Ritchie at Bell Labs.

**Callback function** – Code that's called, or run, when a specific event happens.

**Calling a function** – The phrase programmers use to mean they are using a function.

**capitalize() function** – A string method that makes the first letter in a string uppercase.

**Case** – The case of a letter is whether a letter is lowercase or uppercase.

**Character** – The smallest unit in a written language.

**circle()** – In the turtle program, the circle() command makes the turtle draw a perfect circle.

**Citation** – When you officially give credit to the person who created a work. This might be an author or an artist, for example.

**Class** – An object that can be used to make copies of itself. See also **instance**.

**Click and drag event** – When the user clicks the mouse button and continues pressing the mouse button while moving the mouse.

**Click event** – An event created by a user clicking a mouse button.

**Client** – The person or group of people who hire a programmer or programmers to develop and write a program.



© Copyright 2010 Giant Campus

**Client requirements** – All of the things that a client needs a completed software program to do.

**Closed source software** – Software that hides its code and doesn't let other people see it. Closed source software is also called proprietary software.

**Code** – Text that is written in a human-readable programming language that a computer can also interpret. Code is sometimes referred to as source code.

**Code block** – See **block of code**.

**Code complete** – A software program is called code complete when no code will be written in order to add new features. New code may be written in order to fix bugs.

**Code completion** – See **autocomplete**.

**Code style** – How program code looks and what kinds of comments, names, and layout a programmer uses when writing it.

**color()** – In the turtle program, the color() command lets you change the turtle's color, which not only changes the color of the turtle, but the lines it draws as well.

**Command line** – A type of computer interface that uses text-based commands to tell a computer to do something.

**Command prompt** – The main feature of the Python (command line) interpreter. The command prompt is the command line where a programmer types commands. It looks like this: >>>

**Commenting out code** – Using the # symbol to make Python ignore parts of your code. You can comment out lines of code when you want to make Python run the program without that part of the code, but you don't want to delete the code completely.

**Comments** – Notes that programmers use to explain what some code is doing. These comments are meant to be read by the programmer and by anyone else reading the code.

**Compression** – To make something smaller by squishing or squeezing it.

**Computer program** – A set of instructions for a computer made up of one or more lines of code.

**Computer programming** – The process of writing instructions for a computer expressed in code.

**Concatenation** – Adding strings together to make a new string.

**Constraint** – Something that limits a project in some way.

**Control flow** – A programming technique that changes the order in which the lines of a program are run, but it doesn't change the order of the lines themselves.



© Copyright 2010 Giant Campus

**Coordinates** – Numbers that tell you where a point is on the X axis and Y axis. The X coordinate is how far left or right a point is along the X axis. The Y coordinate is how far up or down a point is along the Y axis. At the origin point in the center, the X and Y coordinates are both 0.

**count() function** – A string class method that counts the number of times a substring appears in a string. In the list method, the count() function looks for a specific element in a list and then returns how many times that element is found.

**Cursor** – A symbol that shows you where you are about to type. The cursor usually looks like a blinking line, but it can also look like a blinking square.

**Cut and paste** – One way to move text from one place to another.

**Cut command** – A command that copies text and allows you to paste it somewhere else. Unlike the Copy command, the Cut command deletes the original copy of the selected text.

**Data** – In computer science, data is how a computer represents information. See also **value**.

**Data type** – What kind of value something is.

**Deadline** – A specific date or time that a project must be finished by.

**Debugging** – Fixing errors in a computer program.

**Declarative programming** – Programming that focuses on telling the computer what to do, instead of telling the computer how to do it.

**Declaring a variable** – Creating a variable.

**def** – The command used to define functions.

**Dependency** – A software program or piece of code that a program needs in order to work.

**Design** – In the waterfall design stage, developers use the information gathered in the requirements stage to come up with a solution, and then create a plan, or a design, based on that solution.

**Designer** – A person who helps create the plan for a software project.

**Developer** – See **programmer**.

**Directory** – Another name for a folder. A **parent directory** is the name of a folder that's holding a nested folder. A **child directory** is the name for a folder inside a parent directory along a file path.

**Docstrings** – Special comments used to describe functions, modules, and other Python objects. Docstrings are also called document strings.

**Element** – One object in a list.

**End-users** – The expected users of a software program.



© Copyright 2010 Giant Campus

**Escape character** – Another term for the backslash \ character.

**Escaping a character** – Adding a backlash before a character to create an escape sequence.

**Event** – Something that happens to or on a computer.

**Event-driven programming** – A programming paradigm in which the activity of a program is controlled by events.

**Event handler** – Code in a program that waits for a specific event, such as a mouse click, and then does something in response to the event, such as drawing a shape.

**Execute** – Another word for running a program.

**exit()** – In the turtle program, the exit() command will close the Python (command line) window as soon as the ENTER key is pressed.

**exitonclick()** – In the turtle program, the exitonclick() command keeps the turtle graphics window open until someone clicks it with a mouse.

**Explorer window** – Another name for the Windows Explorer window.

**Expression** – A programming statement that returns a new value.

**External factors** – Things outside the control of the software development team.

**Extreme Programming (XP)** – A type of agile software development which emphasizes the traditional programming practices that are considered by most programmers as good practices, while de-emphasizing those practices which are considered problematic.

**Feature complete** – The name for when all of the planned features of a software program have been added.

**Feature Driven Development (FDD)** – A process in which programming teams and large pieces of code are organized according to the features a program should have.

**File compression (zip) utility** – A program that turns regular files into compressed files.

**File manager program** – An interface that lets a user work with files on a computer. File manager programs are also called file managers.

**File name extensions** – A special set of letters or numbers at the end of a file name that show what type of file it is.

**File path** – A way to refer to a file's specific location on a computer's hard drive. Like a website's URL, anyone with access to a specific computer can use a file path to access a file.

**Finder** – The file manager program used by Apple's Mac operating systems.

**First-class functions** – Functions that send functions as arguments to other functions, receive functions as values from other functions, and assign functions as variables.



© Copyright 2010 Giant Campus

**First-class objects** – Objects created while a program runs.

**Floating point number** – A number that has a decimal point.

**Flowcharts** – Diagrams which try to represent all the steps in a process.

**Focus** – Making a window active.

**Folder view** – A way of viewing the files and folders on your computer.

**Forking** – Taking code from a currently developed project in order to make a different version.

**for loop** – A loop that iterates through a block of code a specific number of times or until something specific happens in the program.

**Formal development method** – A software development model that that uses and follows a strict and predictable order for when specific development actions take place.

**forward()** – In the turtle program, forward() is a command which moves the turtle forward in the direction its currently facing.

**Forward slash** – The / character.

**Freezing** – When a window stops responding to a user.

**from turtle import*** – Code that tells Python to load a program called turtle.

**Function** – A set of code that's been given a specific name. A programmer can type a function name every time they want to use that function's code, instead of reusing all of the code again.

**Functional programming** – A paradigm that emphasizes using functions instead of changing the state of a program, like a procedural program would do.

**Functional requirements** – The things a software program needs to do when it's completed.

**Functional specification** – Part of a plan that describes what the software program should do when users interact with it. It does not describe the software's code or internal processes.

**goto() command** – In the turtle program, the goto() command uses X and Y coordinates to move the turtle to a specific point in the turtle graphics window.

**Graphical User Interface (GUI)** – All of the windows, buttons, menu items, icons, and dialog boxes the computer user can interact with.

**Hardware** – All the physical parts of a computer that can be seen and touched, such as wires, buttons, circuits, CD-ROM drives, and other physical parts.

**help() function** – A function built into Python that lets you find out more information about Python's built-in functions, syntax, and other Python-related topics, such as Python classes and modules.



© Copyright 2010 Giant Campus

**Home directory** – A directory assigned to a specific user as a place for storing all of that user's files.

**Horizontal line** – A line that goes straight across from side to side. See also **x axis**.

**IDE** – See **integrated development environment**.

**IDLE** – A software program, written in Python, that is used to write and run code and programs written in Python.

**Immutable** – See **mutability**.

**Imperative programming** – See **procedural programming**.

**Implementation** – In the waterfall implementation stage, developers use the plan created in the design stage to write code.

**Incremental** – To do something in small steps or chunks.

**Index** – In a book, an index is a list of words or phrases used in the book and the page numbers where those terms can be found in the book.

**index() function** – A string class method that finds the location of the first time a substring is used in a string, and then returns the index of that substring. The index() function is also a list method that finds the index of the first element in a list that matches the index() function's argument.

**Index of a character** – A number that tells you where the character is located in a string.

**Informal development method** – A development method in which the development does not follow a predictable or planned course of action.

**Input requirements** – How a program needs to receive information from its users or other sources.

**insert() function** – A function that adds a new element to a list by putting it into the list at a specific index.

**Instance** – Each copy of a class.

**Integer** – A positive or negative number without a fraction or decimal point. Zero is also an integer.

**Integrated development environment** – A type of software program used to write code and develop other programs. An integrated development environment is also referred to as an IDE.

**Interaction** – An interaction between a user and a computer happens when a computer receives input from the user or provides output to the user.

**Interactivity** – The way that people use computers, as well as the ways in which computers respond to what people do.



© Copyright 2010 Giant Campus

**Interface** – A specific method for interacting with a computer operating system or program.

**Interpreter** – A program that runs the code a programmer writes.

**Iteration** – One cycle of a loop.

**Iterative and incremental models** – Models of software development that are based on repetition and many small changes.

**Iterator-based for loop** – A loop that ends when it finishes iterating through a list of numbers.

**Iterator variable** – A temporary variable that can be used to keep track of how many times the for loop has repeated.

**Java** – A programming language that was developed in 1995 by James Gosling at Sun Microsystems.

**Java applets** – Java programs on the web.

**join() function** – A string method that can convert list elements into a string.

**Keywords** – Python programming terms that have specific meanings in Python.

**left()** – In the turtle program, the left() command rotates the turtle counterclockwise from the direction it's facing.

**len() function** – A method for the class list. It tells you the number of elements in a list.

**letter_height** – A variable that's assigned an integer value that sets the height of the drawn letters.

**letter_width** – A variable that's assigned an integer value that sets the width of the drawn letters.

**List** – A series of objects. A list can be made up of strings, integers, combinations of objects of different data types, and even other lists.

**listen()** – A function that makes the Python Turtle Graphics window the active window in order to detect, or listen for, specific events, such as mouse clicks or specific key presses.

**Literal** – When something is what it appears to be, without exaggeration or difference. In programming, a literal is also used to mean an object that is what it appears to be.

**Logo** – A programming language that was created in 1967 for educational uses.

**Loop** – A block of code that repeats a specific number of times. Looping is also called iterating. See also **iteration**.

**lower() function** – A function that makes all of the characters in a string lowercase.



© Copyright 2010 Giant Campus

**Main hard drive** – On a computer, this is the place where most of the files and folders are stored. It's usually called the C: drive, but some computers will have other names for it.

**Maintenance** – In the waterfall maintenance stage, developers add new code or fix broken code based on the tests in the verification stage.

**Manipulation** – When you change a value.

**Metaprogramming** – Writing a program that writes or changes other programs.

**Methods** – The functions associated with a class.

**Model** – A re-creation of a system designed to show how that system works. In software development, models are methods used to plan how the software development process will work before it starts and to organize the way people work while developing the software.

**Modular code** – Code that can be reused in different places.

**Module** – A Python program that can be imported.

**Mutability** – A way of describing the kinds of changes you can make to a value. Some data types are mutable, and others aren't. An immutable data type is a value that can't be changed after it has been created. A mutable value data type is a value that can be changed after it has been created.

**Mutable** – See **mutability**.

**Navigate** – When you open folders to get to an address.

**Nested folders** – Folders kept inside other folders.

**Newline** – A special character or sequence of characters that a computer interprets as the end of a line of text and the start of a new one.

**Non-alphanumeric character** – A character that is not either a letter or a number.

**Non-functional requirements** – Another word for constraints.

**Object** – Anything that can be used by Python to do something. Objects are values, variables, functions, and many other things.

**Object-oriented programming** – A programming paradigm in which programs are made up of objects which interact with one another.

**Objectives** – A project's requirements and goals.

**onclick()** – A special type of function used in the turtle program to register callback functions to specific click events.

**ondrag()** – A function that registers the callback function that should be used when the click and drag event happens.



© Copyright 2010 Giant Campus

**onkey()** – A function that links a keyboard key press to a specific function.

**Open source software** – Software that makes its code available for anyone to look at and change.

**Operating system** – Manages all of the hardware and software on a computer. It helps the physical parts of a computer communicate with each other and makes sure the software programs can complete tasks.

**Operators** – Characters that programming languages use to do math. The **+ operator** does addition. The **- operator** does subtraction. The *** operator** does multiplication. The **/ operator** does division.

**Origin point** – The point where the x axis and y axis intersect, in the center of the turtle graphics window.

**Output requirements** – How a program needs to display its results or share them with a user through a file or a print-out.

**Parsing errors** – Errors in the syntax of a program's code.

**pd()** – See **pendown()**.

**pen_color** – The variable that will be assigned a string value used to set the turtle's pen color. This will change the color of the lines the turtle draws.

**pen_width** – The variable that will be assigned an integer value used to set the width of the pen lines.

**pendown()** – In the turtle program, after you've used the penup() command, you can use the pendown() command to make the turtle start drawing again.

**penup()** – In the turtle program, the penup() command lets you move the turtle without drawing a line. The penup() command can be abbreviated to pu()

**Perl** – A programming language that was developed in the mid 1980s by programmer Larry Wall.

**Pixels** – The tiny colored dots that make up the words and images you see on your computer screen.

**Plagiarism** – When one person represents someone else's work as their own.

**pop() function** – A list method that deletes an element at a specific index in the list. While deleting it, it also returns that list element value.

**Pre-alpha release** – The software development phase leading up to the alpha release. Pre-alpha can include all of the previous stages of the software development process.

**Principles** – Generalizations about how to write code which are meant to guide the programmer through making good coding decisions.



© Copyright 2010 Giant Campus

**print() function** – A function used to output strings, numbers, or variable values to the screen.

**Procedural programming** – A programming paradigm uses a list of instructions called procedures to complete its tasks. Procedural programming is sometimes called imperative programming.

**Programmer** – A person who writes code. A programmer is sometimes called a software developer or developer.

**Programming paradigm** – A way of structuring how the code is written and how the pieces of the code interact with each other.

**Project manager** – The person who is in charge of a team working on a software project.

**Proprietary software** – See **closed source software**.

**pu()** – See **penup()**.

**Python** – A programming language used to write software programs.

**Python (command line) interpreter** – A program that creates a window where a programmer can enter and run Python commands.

**Quality Assurance (QA)** – Testing that focuses on how well a program works, and not whether it works.

**radius()** – The distance from the center of a circle to the line of the circle.

**range()** – A function that uses integers to generate a list of numbers. The range() function accepts many different types of arguments, but the simplest argument is a single integer.

**Raw strings** – Strings in which Python doesn't read backslashes as escape characters. In a raw string, backslashes are interpreted as literal backslashes, and not escape characters.

**Reflective programming** – A paradigm in which a program can track and change its structure and behavior as it runs.

**Registering a callback function** – The name for connecting a function to a specific user event, such as a click event.

**Release** – The name for distributing software to the public or to the client who will be using the software.

**Release candidate** – The name for a software version that has the possibility of being ready to distribute to regular users of the software.

**remove() function** – A list method that searches for a specific element, and then deletes the first one it finds.

**replace() function** – A string method that finds all of the substrings that match a specific character or set of characters, and then replaces them with another character or set of characters.



© Copyright 2010 Giant Campus

**Requirement** – A necessary and specific part of a solution that you can measure or test to see if your software design has solved the problem.

**Requirements** – In the waterfall requirements stage, software developers gather all the information they need about the problem they're trying to solve, and then collect that information together in one place.

**reset()** – In the Turtle program, the reset() command erases everything drawn by the turtle and moves the turtle back to its starting position.

**Returning** – When a function sends a value to another function for it to use, assigns a value to a variable, or outputs a value.

**Reversing** – The technique of taking a string and reversing it, by making the first character in the string last and the last character first, and swapping the positions of all the characters in between.

**right()** – In the turtle program, the right() command rotates the turtle clockwise from the direction it's facing.

**Risks** – Constraints on a program's development.

**Root directory** – On Mac and Unix operating systems, the root directory is the highest-level directory on a computer system. All of the directories and files on a computer are inside the root directory.

**Ruby** – A programming language that was released in 1995 by creator Yukihiro Matsumoto.

**Scope of work document** – A document like the software design document that's intended to communicate the software plan to the person or group of people who have requested the software solution. A scope of work is sometimes called a statement of work document.

**shape()** – In the Turtle program, a command that changes the shape of the turtle.

**Shelfware** – Work that someone does that ends up not being read or used in an actual development process.

**Shell** – A software interface that stands between the user and the lowest-level software program of a computer system, making it easier for the user to work with that computer system.

**Shortcut** – A small file that contains the full file path to another file or folder. A shortcut points to another file or folder. In Mac OS X, shortcuts are called aliases.

**Simple model** – A model of software development which follows the following steps:

- Define the problem that needs to be solved
- Think of possible ways to solve the problem
- Select a solution form the problem
- Create a detail plan for the software program
- Write the code for the software program
- Test the program to see how well it works.



© Copyright 2010 Giant Campus

**Slicing** – Grabbing a substring out of a string.

**Slipping** – When one part of a software project has more errors than the others, which keeps the other parts of the project from being released.

**Software** – The computer's non-physical parts. It can't be seen or touched, but it can display things on a computer screen. Software can do things like store information and help the physical parts of the computer communicate with each other.

**Software architect** – A person who plans, designs, and oversees the construction of software projects.

**Software design document** – A description of how to create a software program. The software design document is used to coordinate the activity of a software development team. It outlines all the parts of the software program and how they will work together.

**Software developer** – See **programmer**.

**Software development process** – The way in which a software program is designed, created, and maintained.

**Software documentation** – A description of how to use a software program in its entirety.

**Software package** – A module which can be used by a programming language.

**Software program** – A type of software with a specific purpose, such as sending and receiving email, creating text documents, browsing the web, and other tasks. You're using a software program right now to view this course.

**Software Requirements Specification (SRS)** – The specific and formal documentation of a software project's requirements, constraints, and goals.

**sort() function** – A list method that sorts the elements in a list.

**Source code** – See **code**.

**space_width** – A variable that’s assigned an integer value that sets the length of the space between letters.

**Special character** – A character with a specific meaning attached to it in addition to what the character appears to be. In Python, special characters are characters that have a special meaning when the character is used immediately after the backslash \ character.

**Spiral model** – A model of software development that is one type of iterative and incremental development. The spiral model has the following stages:

- Determine objectives
- Identify and resolve risks
- Write and test the code
- Plan the next iteration

**split() function** – A string method you can use to split a string into a list.



© Copyright 2010 Giant Campus

**Stakeholder** – Any participant in the software project being developed or anyone who needs the software project to be completed for some reason.

**Statement** – The smallest piece of code that can still do something.

**Statement of work document** – See also **scope of work document**.

**Steps** – In the circle() command, the steps argument is the number of sides, or steps, the turtle should use to draw a circle.

**Stride number** – An integer that determines how large the strides should be and which characters should be kept from the sliced string. Each character counts as a stride size of 1.

**Striding** – In normal language, striding means walking. Walking is made up of specific individual steps called strides. In programming, striding is an operation that moves through the indexes of the characters in a string by using strides of a specific size.

**String** – A sequence of one or more characters with apostrophes or quotation marks around them.

**String literals** – Strings that are used directly in Python code, instead of using a string assigned to a variable.

**String manipulation** – A group of computer programming techniques used to work with strings.

**String processing** – See **string manipulation**.

**Structure** – The way code is organized and the coding methods used to solve particular problems.

**Style guides** – Documents which provide more details about how specific types of code should be written.

**Substring** – A character or set of characters inside a string.

**swapcase() function** – A function that makes all the lowercase characters in a string uppercase and all of the uppercase characters in a string lowercase.

**Syntax** – The set of rules for how a programming language needs to be written. Syntax is like the grammar of a programming language.

**System processing requirements** – What a program needs to do with the input it receives.

**Taking arguments** – When a function uses optional or required arguments when the function is called.

**Taskbar** – An area that shows you all of the things you have open on your computer.

**Test case** – A description of a specific user activity and what the outcome needs to be for the software program to pass the test.



© Copyright 2010 Giant Campus

**Tester** – A person who tests a software program to make sure it works.

**Text editor** – IDLE's tool for writing, saving, and running Python programs.

**Topic** – A concept about programming in Python.

**Unassigned variable** – A variable without a value.

**undo()** – In the turtle program, the undo() command will undo the last turtle command that you entered. You'll use the undo() command when you make a mistake and want to undo it. This is different from the Python text editor's **Undo command**.

**Undo command** – A command in Python's text editor that is used to undo mistakes typed into the text editor. This is different from the turtle program's **undo()** command.

**Unicode** – The standard system for assigning numeric values to represent the many letters and other characters that people want to use to interact with computers.

**Unidirectional** – Moving in a single direction. See also **bidirectional**.

**Unzipping** – When you return a zipped file to its normal size. Unzipping a file is also called extracting it.

**upper() function** – A function that makes all of the characters in a string uppercase.

**Usability** – A measure of how easy to use something is.

**Usability testing** – A way of making sure that the program works the way an end-user expects it to. Usability testing usually focuses on the user interface over more internal systems.

**Use cases** – Specific actions that a user might perform when using the software program.

**User Centered Design (UCD)** – A design philosophy that uncovers possible solutions to a problem by focusing on the people affected by that problem.

**User created event** – An event that happens when a user provides input to a computer.

**Value** – A single piece of information or a collection of information grouped together in one place. Values are sometimes called data.

**Variables** – Names given to specific values. Variables are like labels or name tags that you put on values.

**Verification** – In the waterfall verification stage, developers test the code that was written in the implementation phase.

**Version control systems** – Software programs which keep track of any changes made to a software program's code over time. They also keep track of who made changes to the code.

**Vertical line** – A line that goes straight up and down. See also **y axis**.



© Copyright 2010 Giant Campus

**Visual Basic** – A programming language developed by Microsoft.

**Waterfall model** – A development methodology in which each phase of the design process leads to the next, flowing downward like a waterfall.

**Web browser** – A software program for viewing and using web pages.

**Whitespace** – Space in a text editor that has no characters in it. It's usually created with spaces from the SPACEBAR or with tabs from the TAB key.

**Window** – A square shape that contains software program open in an operating system.

**Windows Explorer** – Microsoft's file manager program.

**X axis** – An invisible horizontal line that goes across the center of the turtle graphics window. See also **horizontal line** and **y axis**.

**X coordinate** – See **coordinate**.

**Y axis** – An invisible vertical line that goes straight through the center of the turtle graphics window. See also **vertical line** and **x axis**.

**Y coordinate** – See **coordinate**.

**Zipped file** – A compressed file. It's a file that has been made smaller than it was originally.



© Copyright 2010 Giant Campus