

Chapter 1 : Basic Java Examples

As displayed in the above diagram, write the simple program of java in notepad and saved it as calendrierdelascience.com To compile and run this program, you need to open the command prompt by start menu -> All Programs -> Accessories -> command prompt.

Hello World In this section, our plan is to lead you into the world of Java programming by taking you through the three basic steps required to get a simple program running. As with any application, you need to be sure that Java is properly installed on your computer. You also need an editor and a terminal application. Here are system specific instructions for three popular home operating systems. We break the process of programming in Java into three steps: Create the program by typing it into a text editor and saving it to a file named, say, MyProgram. Compile it by typing "javac MyProgram. Execute or run it by typing "java MyProgram" in the terminal window. The first step creates the program; the second translates it into a language more suitable for machine execution and puts the result in a file named MyProgram. Creating a Java program. A program is nothing more than a sequence of characters, like a sentence, a paragraph, or a poem. To create one, we need only define that sequence characters using a text editor in the same way as we do for email. Type these character into your text editor and save it into a file named HelloWorld. A compiler is an application that translates programs from the Java language to a language more suitable for executing on the computer. It takes a text file with the. Otherwise, go back and make sure you typed in the program exactly as it appears above. Executing or running a Java program. Once you compile your program, you can execute it. This is the exciting part, where the computer follows your instructions. To run the HelloWorld program, type the following in the terminal window: The key line with System. When we begin to write more complicated programs, we will discuss the meaning of public, class, main, String[], args, System. Creating your own Java program. For the time being, all of our programs will be just like HelloWorld. The easiest way to write such a program is to: Replace HelloWorld with the program name everywhere. Replace the print statement by a sequence of statements. Most errors are easily fixed by carefully examining the program as we create it, in just the same way as we fix spelling and grammatical errors when we type an e-mail message. These errors are caught by the system when we compile the program, because they prevent the compiler from doing the translation so it issues an error message that tries to explain why. These errors are caught by the system when we execute the program, because the program tries to perform an invalid operation e. These errors are hopefully caught by the programmer when we execute the program and it produces the wrong answer. They can be subtle and very hard to find. One of the very first skills that you will learn is to identify errors; one of the next will be to be sufficiently careful when coding to avoid many of them. Typically, we want to provide input to our programs: The simplest way to provide input data is illustrated in UseArgument. Whenever this program is executed, it reads the command-line argument that you type after the program name and prints it back out to the terminal as part of the message. Exercises Write a program TenHelloWorlds. Web Exercises Write a program Initials.

Chapter 2 : Java programs | Programming Simplified

java is cross platform programming language, it means you can write your program in windows, linux, mac or any other operating system and it will also runs in any OS as long as you do have JDK. java is cover almost all application in computer from web, application, games and many more.

Compiled code of a program should be executed in any operating system, irrespective of the as in OS in which that code had been generated. This concept is known as platform independence. The birth of oops concept took place with encapsulation. Any program contains two parts. Data part and Logic part Out of data and logic the highest priority we have given to data. But in a structured programming language, the data insecurity is high. Thus in a process, if securing data in structured prog. In structured programming lang programs, the global variable play a vital role. But because of these global variables, there is data insecurity in the structured programming lang programs. In this way data is unsecured. But this is not the actual reason. Assume that it is a project. Now if any upgradation is required, then the client i. Now we should note that it is not guaranteed that the programmers who developed this program will still be working with that company. Hence this project falls into the hands of new programmers. Automatically it takes a lot of time to study. The project itself before upgrading it. It may not be surprising that the time required for writing the code to upgrade the project may be very less when compared to the time required for studying the project. Thus maintenance becomes a problem. If the new programmer adds a new function to the existing code in the way of upgrading it, there is no guarantee that it will not affect the existing functions in the code. This is because of global variables. In this way, data insecurity is created. To overcome this problem, programmers developed the concept of encapsulation. For example, let us have a struc. It is sure that all the twenty functions will not use all the global variables. Three of the global variables may be used only by two functions. But in a structured prog. Every function will have access to all the global variables. To avoid this problem, programmers have designed a way such that the variables and the functions which are associated with or operate on those variables are enclosed in a block and that bock is called a class and that class and that class is given a name, Just as a function is given a name. Now the variables inside the block cannot be called as the local variable because they cannot be called as global variables because they are confined to a block and not global. Hence these variables are known as instance variables Ex:

Chapter 3 : Introduction to Java programming, Part 1: Java language basics

Hi Chaitanya, thanks a lot for such a useful website for those of us who are just starting to learn Java! The basic programs are very helpful. Reply. Naveen says.

To compile the program, type: A footprint model simply defines the area that will be covered by an object. Footprint models of buildings are commonly used in the earliest stages of design and in high-level models of urban areas. Because the footprint area is defined by its perimeter, naturally, a general-purpose polygon model is the first approach that comes to mind. However, it turns out that the computation of simple polygon operations e. Many potentially difficult problems can be avoided by modeling the footprint as a collection of simple triangular regions. The next figure shows, for example, a six-triangle footprint model for the AV. Class diagram for footprint implementation Simply put, the adjacent figure says that one Footprint object will be composed of many Triangle objects. In turn, triangles will be defined in terms of Node and Edge objects. Nodes are an extension of Vector. Properties of the building footprint e. Download the zipped source code. Model of a Wheel Cross-Section. In this example we assemble a cross-section model of a wheel from triangle objects, and display the wheel cross section and its area in a graphics window. Pixel coordinates as measured from the top left-hand corner of the graphics canvas can also be printed on the canvas by moving the cursor to a desired position and then clicking. Cross section model of a Wheel. In this specific example, the minimum radius is 50 and the maximum radius is Coordinates in the radial and angular directions are divided into three and sixteen intervals, respectively. Theoretical considerations indicate that the cross section will be 62, However, because the triangles only approximate the circular cross section, the computed area will be a little less. Individual triangle objects are assembled from Node and Edge objects, as shown above in the Footprint model. The wheel cross section is simply a collection of triangles stored in an array list. The mesh of triangles in generated with a double loop. One loop increments the coordinates from the minimum radius to the maximum radius. A second loop systematically generates nodal coordinates in an angular direction, from 0. This program computes the engineering properties cross section area; x,y coordinates of the centroid; moments of inertia about the centroid for a polygon cross section, possibly containing holes. Screenshot of bridge cross section and engineering properties Polygon cross sections are modeled as an exterior ring, possibly containing multiple interior rings. Each ring corresponds to a list of nodes have x,y coordinates. The nodes in an exterior ring are stored in a clockwise direction. Interior rings holes have nodes stored in an anti-clockwise direction. Polygon cross sections are defined in an xml file. Rather than put individual java and xml files online, I have bundled them into a tar file called BridgeAnalysis. To unpack the tar file, type: Move to that directory and then compile the program with the command: A day planner is simple graphical program for presenting a calendar of events. For example, Figure Screenshot of Day Planner application. The schedule of events in stored in an XML file called planner. Here is an abbreviated snippet of code: Modeling of a Simple Transportation Network. Networks are a fundamental aspect of Civil Engineering Systems. For example, transportation engineers are concerned with the design and operation of rail, road, shipping, and air traffic networks. Hydrology engineers work with networks of waterways. Structural engineers work with networks of load bearing members usually beams and columns. The following figure is a simplified model of the Washington DC metro system most of the stations have been omitted: Simplified view of the Washington DC Metro System Passengers need to be familiar with the routes, schedule, and cost of using the metro. When will the metro open? When will it close? How do I get from station A to station B? Will I have to change trains? How long will it take to get there? And how much will it cost? When is the last train to station B? The Smithsonian question is particularly interesting because: In its most rudimentary form, the Metro System can be modeled as a graph structure. Nodes in the graph correspond to train stations; the edges correspond to the tracks connecting stations. Graph theory is useful because it provides algorithms to answer questions about routes connecting stations A and B. The adjacent code is a first-cut implementation at modeling this problem domain. Question How would you extend the basic node-edge graph structure to make the model more realistic. A fundamental problem in Transportation Engineering is the planning of routes for expansion of

transportation networks. Suppose, for example, that we want to build a road from city A to city B, but that a mountain range spans the most direct route. Is it better to build a road around the mountains, or pay more money upfront to build a tunnel through the mountains and provide a shorter route? The standard approach to problems of this type is to deal with each concern separately, and then combine the results. For example, the physical constraints might look like: Dealing with physical constraints in the route selection problem The final result is always never a single point, but rather a family of good solutions: Pareto Optimal Curves Program

Chapter 4 : Java Tutorial - Basic Syntax and Program, Simple Program, Java Simple example, First java p

Java Programs or Java programming tutorial with examples of fibonacci series, armstrong number, prime number, palindrome number, factorial number, bubble sort, selection sort, insertion sort, swapping numbers etc.

Java - Package Java - Basic Program When we consider a Java program it can be defined as a collection of objects that communicate via invoking each others methods. Let us now briefly look into what do class, object, methods and instance variables mean. Object - Objects have states and behaviors. A dog has states-color, name, breed as well as behaviors -wagging, barking, eating. An object is an instance of a class. Methods - A method is basically a behavior. A class can contain many methods. It is in methods where the logics are written, data is manipulated and all the actions are executed. Instance Variables - Each object has its unique set of instance variables. Let us look at a simple code that would print the words Simple Test Promgram. Please follow the steps given below: Open notepad and add the code as above. Save the file as: Open a command prompt window and go o the directory where you saved the class. If there are no errors in your code the command prompt will take you to the next line. The path variable is set. The public keyword is an access specifier, which allows the programmer to control the visibility of class members. When a class member is preceded by public, then that member may be accessed by code outside the class in which it is declared. In this case, main must be declared as public, since it must be called by code outside of its class when the program is started. The keyword static allows main to be called without having to instantiate a particular instance of the class. This is necessary since main is called by the Java interpreter before any objects are made. The keyword void simply tells the compiler that main does not return a value. As you will see, methods may also return values. As stated, main is the method called when a Java application begins. Keep in mind that Java is case-sensitive. Thus, Main is different from main. It is important to understand that the Java compiler will compile classes that do not contain a main method. But the Java interpreter has no way to run these classes. So, if you had typed Main instead of main, the compiler would still compile your program. However, the Java interpreter would report an error because it would be unable to find the main method. Any information that you need to pass to a method is received by variables specified within the set of parentheses that follow the name of the method. These variables are called parameters. If there are no parameters required for a given method, you still need to include the empty parentheses. In main , there is only one parameter, albeit a complicated one. String args[] declares a parameter named args, which is an array of instances of the class String. Objects of type String store character strings. In this case, args receives any command-line arguments present when the program is executed.

Chapter 5 : 4 Ways to Write Your First Program in Java - wikiHow

Java programming: Java program code consists of instructions which will be executed on your computer system to perform a task as an example say arrange given integers in ascending order. This page contains sample programs for beginners to understand how to use Java programming to write simple Java programs.

August 24, Content series: This content is part of in the series: Introduction to Java programming, Part 1 [http:](http://) This content is part of the series: Introduction to Java programming, Part 1 Stay tuned for additional content in this series. Find out what to expect from this tutorial and how to get the most out of it. About this tutorial The two-part Introduction to Java programming tutorial is meant for software developers who are new to Java technology. Work through both parts to get up and running with object-oriented programming OOP and real-world application development using the Java language and platform. This first part is a step-by-step introduction to OOP using the Java language. Programming examples in Part 2 build on the Person object that you begin developing in Part 1. Follow up with " Introduction to Java programming, Part 2: Constructs for real-world applications " to build on this foundation. Prerequisites This tutorial is for software developers who are not yet experienced with Java code or the Java platform. The tutorial includes an overview of OOP concepts. System requirements To complete the exercises in this tutorial, you will install and set up a development environment consisting of: The recommended system configuration is: At least MB of disk space to install the software components and examples. Java platform overview Java technology is used to develop applications for a wide range of environments, from consumer devices to heterogeneous enterprise systems. In this section, get a high-level view of the Java platform and its components. By default, you see three panes in the Javadoc. The top-left pane shows all of the packages in the API, and the bottom-left pane shows the classes in each package. The main pane to the right shows details for the currently selected package or class. For example, if you click the java. Like any programming language, the Java language has its own structure, syntax rules, and programming paradigm. Structurally, the Java language starts with packages. Within packages are classes, and within classes are methods, variables, constants, and more. You learn about the parts of the Java language in this tutorial. The Java compiler When you program for the Java platform, you write source code in. Bytecode is a set of instructions targeted to run on a Java virtual machine JVM. In adding this level of abstraction, the Java compiler differs from other language compilers, which write out instructions suitable for the CPU chipset the program will run on. The difference is that the JVM is a piece of software written specifically for a particular platform. Your code can run on any chipset for which a suitable JVM implementation is available. JVMs are available for major platforms like Linux and Windows, and subsets of the Java language have been implemented in JVMs for mobile phones and hobbyist chips. The garbage collector Rather than forcing you to keep up with memory allocation or use a third-party library to do so , the Java platform provides memory management out of the box. When your Java application creates an object instance at runtime, the JVM automatically allocates memory space for that object from the heapâ€” a pool of memory set aside for your program to use. The Java garbage collector runs in the background, keeping track of which objects the application no longer needs and reclaiming memory from them. Garbage collection is one of the essential features of Java platform performance. The Java Development Kit When you download a Java Development Kit JDK , you get â€” in addition to the compiler and other tools â€” a complete class library of prebuilt utilities that help you accomplish most common application-development tasks. The JRE is available for multiple platforms. If you already have the JDK and Eclipse IDE installed, you might want to skip to the " Getting started with Eclipse " section or to the one after that, " Object-oriented programming concepts. Although you can use these tools to develop your applications, most developers appreciate the additional functionality, task management, and visual interface of an IDE. Eclipse is a popular open source IDE for Java development. Eclipse handles basic tasks, such as code compilation and debugging, so that you can focus on writing and testing code. In addition, you can use Eclipse to organize source code files into projects, compile and test those projects, and store project files in any number of source repositories. If you download one of the Eclipse bundles, it will come with the JDK already. Agree to the license terms for the version you want to

download. Choose the download that matches your operating system and chip architecture. Windows Save the file to your hard drive when prompted. When the download is complete, run the install program. Install the JDK to your hard drive in an easy-to-remember location such as C: OS X When the download is complete, double-click it to mount it. Run the install program. You do not get to choose where the JDK is installed. You now have a Java environment on your computer. Install Eclipse Browse to the Eclipse packages downloads page. Under Download Links on the right side, choose your platform the site might already have sniffed out your OS type. Click the mirror you want to download from; then, save the file to your hard drive. When the download finishes, open the file and run the installation program, accepting the defaults. To set up your Eclipse development environment: Launch Eclipse from your local hard disk. When asked which workspace you want to open, choose the default. Close the Welcome to Eclipse window. The welcome window is displayed each time you enter a new workspace. You can disable this behavior by deselecting the "Always show Welcome at start up" check box. Figure 1 shows this selection highlighted in the Eclipse setup window for the JRE. Eclipse is now set up and ready for you to create projects, and compile and run Java code. The next section familiarizes you with Eclipse. This section is a brief hands-on introduction to using Eclipse for Java development. The Eclipse development environment The Eclipse development environment has four main components: Workspace Perspectives Views The primary unit of organization in Eclipse is the workspace. A workspace contains all of your projects. A perspective is a way of looking at each project hence the name , and within a perspective are one or more views. Figure 2 shows the Java perspective, which is the default perspective for Eclipse. You see this perspective when you start Eclipse. Eclipse Java perspective View image at full size The Java perspective contains the tools that you need to begin writing Java applications. Each tabbed window shown in Figure 2 is a view for the Java perspective. Package Explorer and Outline are two particularly useful views. The Eclipse environment is highly configurable. Each view is dockable, so you can move it around in the Java perspective and place it where you want it. For now, though, stick with the default perspective and view setup. Create a project Follow these steps to create a new Java project: New Java Project wizard Enter Tutorial as the project name and use the workspace location that you opened when you opened Eclipse. Click Finish to accept the project setup and create the project. You have now created a new Eclipse Java project and source folder. Your development environment is ready for action. Object-oriented programming concepts and principles The Java language is mostly object oriented. This section is an introduction to OOP language concepts, using structured programming as a point of contrast. What is an object? Object-oriented languages follow a different programming pattern from structured programming languages like C and COBOL. The structured-programming paradigm is highly data oriented: You have data structures, and then program instructions act on that data. Object-oriented languages such as the Java language combine data and program instructions into objects. An object is a self-contained entity that contains attributes and behavior, and nothing more. Instead of having a data structure with fields attributes and passing that structure around to all of the program logic that acts on it behavior , in an object-oriented language, data and program logic are combined. This combination can occur at vastly different levels of granularity, from fine-grained objects such as a Number, to coarse-grained objects, such as a FundsTransfer service in a large banking application. Parent and child objects A parent object is one that serves as the structural basis for deriving more-complex child objects. A child object looks like its parent but is more specialized. With the object-oriented paradigm, you can reuse the common attributes and behavior of the parent object, adding to its child objects attributes and behavior that differ. Object communication and coordination Objects talk to other objects by sending messages method calls, in Java parlance.

Chapter 6 : JavaScript Examples

Java, With the help of this course, students can now get a confident to write a basic program to in-depth algorithms in C Programming or Java Programming to understand the basics one must visit the list Java programs to get an idea.

Read on for a primer on OOP concepts in Java. They are an abstraction, encapsulation, inheritance, and polymorphism. Grasping them is key to understanding how Java works. Basically, Java OOP concepts let us create working methods and variables, then re-use all or part of them without compromising security. Abstraction means using simple things to represent complexity. In Java, abstraction means simple things like objects, classes, and variables represent more complex underlying code and data. This is important because it lets avoid repeating the same work multiple times. This is the practice of keeping fields within a class private, then providing access to them via public methods. This way, we can re-use objects like code components or variables without allowing open access to the data system-wide. This is a special feature of Object Oriented Programming in Java. It lets programmers create new classes that share some of the attributes of existing classes. This lets us build on previous work without reinventing the wheel. This Java OOP concept lets programmers use the same word to mean different things in different contexts. One form of polymorphism in Java is method overloading. The other form is method overriding. See more on this below. For example, a programmer can create several different types of objects. These can be variables, functions, or data structures. Programmers can also create different classes of objects. These are ways to define the objects. For instance, a class of variable might be an address. The class might specify that each address object shall have a name, street, city, and zip code. The objects, in this case, might be employee addresses, customer addresses, or supplier addresses. How Encapsulation Works Encapsulation lets us re-use functionality without jeopardizing security. For example, we may create a piece of code that calls specific data from a database. It may be useful to reuse that code with other databases or processes. Encapsulation lets us do that while keeping our original data private. It also lets us alter our original code without breaking it for others who have adopted it in the meantime. It works by letting a new class adopt the properties of another. We call the inheriting class a subclass or a child class. The original class is often called the parent. We use the keyword `extends` to define a new class that inherits properties from an old class. How Polymorphism Works Polymorphism in Java works by using a reference to a parent class to affect an object in the child class. Two more examples of polymorphism in Java are method overriding and method overloading. In method overriding, the child class can use the OOP polymorphism concept to override a method of its parent class. That is, a single method name might work in different ways depending on what arguments are passed to it. Inheritance can be as easy as using the `extends` keyword: The `Employee` class inherits from the `Person` class by using the keyword `extends`. Here, the child class overrides the parent class. For the full example, see this blog post. This is the core concept in Java. You should never have two blocks of identical code in two different places. Instead, have one method you use for different applications. If you expect your Java code to change in the future, encapsulate it by making all variables and methods private at the outset. Simply put, a class should always have only one functionality. Make all methods and classes Closed for modification but Open for an extension. That way, tried and tested code can remain static but can be modified to perform new tasks as needed. For a good, full list of best practices for OOP concepts in Java, see this blog post. You can also check out our article about OOP concepts in C. Latest Posts About Stackify Stackify provides developer teams with unparalleled visibility and insight into application health and behavior, both proactively in a monitoring role as well as reactively in a troubleshooting role, while eliminating the need to login to servers and other resources in order to investigate application problems.

Chapter 7 : What Are OOP Concepts in Java? 4 Primary Concepts

Java Basic Syntax - Learn Java in simple and easy steps starting from basic to advanced concepts with examples including Java Syntax Object Oriented Language, Methods, Overriding, Inheritance, Polymorphism, Interfaces, Packages, Collections, Networking, Multithreading, Generics, Multimedia, Serialization, GUI.

Let us now briefly look into what do class, object, methods, and instance variables mean. A dog has states - color, name, breed as well as behavior such as wagging their tail, barking, eating. An object is an instance of a class. A class can contain many methods. It is in methods where the logics are written, data is manipulated and all the actions are executed. Save the file as: Open a command prompt window and go to the directory where you saved the class. If there are no errors in your code, the command prompt will take you to the next line Assumption: The path variable is set. Java Identifiers All Java components require names. Names used for classes, variables, and methods are called identifiers. In Java, there are several points to remember about identifiers. After the first character, identifiers can have any combination of characters. A key word cannot be used as an identifier. Most importantly, identifiers are case sensitive. Examples of legal identifiers: Examples of illegal identifiers: Java Modifiers Like other languages, it is possible to modify classes, methods, etc. However, an array itself is an object on the heap. We will look into how to declare, construct, and initialize in the upcoming chapters. Java Enums Enums were introduced in Java 5. Enums restrict a variable to have one of only a few predefined values. The values in this enumerated list are called enums. With the use of enums it is possible to reduce the number of bugs in your code. For example, if we consider an application for a fresh juice shop, it would be possible to restrict the glass size to small, medium, and large. This would make sure that it would not allow anyone to order any size other than small, medium, or large. Methods, variables, constructors can be defined inside enums as well. Java Keywords The following list shows the reserved words in Java. These reserved words may not be used as constant or variable or any other identifier names.

Chapter 8 : Java Programs - + Simple Programming Examples & Outputs

If you are new to programming in the Java language, have some experience with other languages, and are familiar with things like displaying text or graphics or performing simple calculations, this tutorial could be for you. It walks through how to use the Java® 2 Platform software to create and run.

Chapter 9 : Java Programming Examples

Java Coding Samples. Various Java programs to illustrate various concepts. A Hello World! Java program. Calling Methods.A sample of how to call methods in the same class.