|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **What is a Subroutine (Method)?**  You may recall in a previous unit, while learning about the **Top Down Problem Solving Approach** in the Program Design stage of the Software Development Process we defined:   |  | | --- | | **Top Down Development**  The intent of this problem solving method is to focus on the broad picture first and concern yourself with the small details later. It is a matter of analyzing a problem and breaking the large problem down into easily manageable blocks. |   The key to this definition is that *a large problem is broken down into easily manageable blocks.* This *divide and conquer* approach to programming allows a problem to be more manageable to solve. In other words, solve the small problems and the large problem solves itself.  A **subroutine** (a subprogram or a sub-task) is a set of programming statements that perform a specific task (or small problem). **Subroutines** are an important part of any programming language, and Java is no exception. In Java subroutines are referred to as **METHODS**. There are generally two types of **methods** in Java:   1. Function Methods: A method that produces a value 2. Procedure Methods: A method that causes one or more actions to occur   Methods (Functions and Procedures) can be:   * provided within the programming language (by SUN Microsystems) * provided by a third party (such as HOLT software) * can be created by the Java programmer.   In all cases methods are created to perform a specific task.  **Advantages of Using Provided Methods** A method that is provided by SUN Microsystems, for example:   |  | | --- | | System.out.println( ); |   or a method provided by a third party such as HOLT Software, for example:   |  | | --- | | age=c.readInt(); |   carry the following advantages:   |  | | --- | | 1. They are provided for the programmer to use without the programmer having to program what is going on behind the scenes. They can save many lines of complex code in the task that they perform. 2. Provided methods are error free. 3. They are easily called upon (used). |   **Advantages of Using Programmer Created Methods** If a programmer uses a modular approach to programming, where each task is programmed as a separate **method,** many advantages arise:   |  | | --- | | 1. Many common data processing methods can be kept in a *personal programming library* for use in numerous programs or for numerous occurances of the same task within one program. In fact, it may even promote method re-use. 2. When you put repeating statements into a method, you decrease your chance of errors because you can carefully debug the code in the method and use it throughout your program. 3. Error detection is easier in small program modules than in one large program. 4. Once you have tested and debugged a method, you can assume it is error free. If errors occur in a program - debugging can be limited to other sections. 5. A "divide and conquer" programming process is an efficient way to solve large problems. In large programs a number of programmers may be used, each solving a small problem. 6. It is easier for another programmer to change a well structure modular program (if change is necessary). |   Visit [Sun Microsystem's Documentation](javascript:na_open_window('win',%20'http://java.sun.com/developer/onlineTraining/JavaIntro/exercises/Methodcall/',%200,%200,%20700,%20450,%200,%201,%200,%201,%201)) for more on 'Defining and Using Java Methods'  [http://angel.peelschools.org/AngelUploads/Content/1-Simler-2672-ICS3U0-A-Fall2010-BEL/_assoc/A241925BEB714CC78951C5E8238B0C7B/images/top212.gif](http://angel.peelschools.org/AngelUploads/Content/1-Simler-2672-ICS3U0-A-Fall2010-BEL/_assoc/A241925BEB714CC78951C5E8238B0C7B/lesson.htm#top) **Top**  http://angel.peelschools.org/AngelUploads/Content/1-Simler-2672-ICS3U0-A-Fall2010-BEL/_assoc/A241925BEB714CC78951C5E8238B0C7B/images/blueline3.gif  **Procedure Methods**  A procedure type method is a method that performs an action or actions when called upon. An example of a procedure method is:   |  | | --- | | c.println("Hello World" ); |   In this example, an action occurs when the procedure **prinln();** is performed. Specifically, the action is displaying the **parameter** (what is found inside of the brackets) on the screen.  **Calling a Method in a Program** To call or invoke a method in a program you must:   * refer to the method's name and specify which parameters are to be sent to the method (these parameters tell the method what to do) * in any call of the method the name of the object to which the method belongs must precede the method's name and is separated from it by a period * in any method called you must send the correct number of parameters   **Graphic Procedure Methods (Part of Holt Software Console Class)**  There a methods that can be used to create graphics as part of the *Console* class. Interesting methods include the  **c.drawRect( )** and the **c.fillRect( )** methods displayed below:     |  |  | | --- | --- | | Sample use of Console Rectangle Methods | In these methods of the console class: The c.drawRect( ) method is used to draw the empty rectangle. The parameters that were passed are the coordinates (in pixels) of the upper left hand corner of the rectangle, the width of the rectangle and the height of the rectangle. The form that the parameters are passed in is: **c.drawRect(x coordinate, y coordinate, width, height)**  In this example the **x coordinate was 25, the y coordinate was 25, the width was 50 and the height was 100** pixels. *NOTE: all values must be integers.*  In these methods of the console class: The c.drawRect( ) method is used to draw the empty rectangle. The parameters that were passed are the coordinates (in pixels) of the upper left hand corner of the rectangle, the width of the rectangle and the height of the rectangle. The form that the parameters are passed in is: **c.drawRect(x coordinate, y coordinate, width, height)**  In this example the **x coordinate was 25, the y coordinate was 25, the width was 50 and the height was 100** pixels. *NOTE: all values must be integers*  Similarly, the c.fillRect( ) method is used to draw a filled rectangle. The default text colour is used to fill in the rectangle with the **x, y, width and height** specified.  In this example the **x coordinate was 100, the y coordinate was 25, the width was 25 and the height was 50** pixels. |   In these methods of the console class: The c.drawRect( ) method is used to draw the empty rectangle. The parameters that were passed are the coordinates (in pixels) of the upper left hand corner of the rectangle, the width of the rectangle and the height of the rectangle. The form that the parameters are passed in is: **c.drawRect(x coordinate, y coordinate, width, height)**  In this example the **x coordinate was 25, the y coordinate was 25, the width was 50 and the height was 100** pixels. *NOTE: all values must be integers*  Similarly, the c.fillRect( ) method is used to draw a filled rectangle. The default text colour is used to fill in the rectangle with the **x, y, width and height** specified.  In this example the **x coordinate was 100, the y coordinate was 25, the width was 25 and the height was 50** pixels.  The console class has numerous methods that can be used to enhance your program. See [the table](javascript:na_open_window('win',%20'console.htm',%2010,%2010,%20650,%20450,%200,%201,%200,%201,%201)) for a partial list.  [http://angel.peelschools.org/AngelUploads/Content/1-Simler-2672-ICS3U0-A-Fall2010-BEL/_assoc/A241925BEB714CC78951C5E8238B0C7B/images/top212.gif](http://angel.peelschools.org/AngelUploads/Content/1-Simler-2672-ICS3U0-A-Fall2010-BEL/_assoc/A241925BEB714CC78951C5E8238B0C7B/lesson.htm#top) **Top**  http://angel.peelschools.org/AngelUploads/Content/1-Simler-2672-ICS3U0-A-Fall2010-BEL/_assoc/A241925BEB714CC78951C5E8238B0C7B/images/blueline3.gif  **Function Methods**  A function type method is a method that produces a value. An example of a function method is:   |  | | --- | | b=Math.sqrt(a); |   In this example, a value (b) is produced by calculating the square root of the parameter (a).  **Calling a Method in a Program** To call or invoke a method in a program you must:   * refer to the method's name and specify which parameters are to be sent to the method (these parameters tell the method what to do) * in any call of the method the name of the object to which the method belongs must precede the method's name and is separated from it by a period (ie. Math) * in any method called you must send the correct number of parameters   **Mathematical Function Methods (Sun Microsystem Math Class - found in the java.lang which is loaded automatically with every Java program)**  There are a number of methods that can be used to perform mathematical functions as part of the *Math* class. Interesting methods include the **sqrt( )** and the **round( )** methods, which are displayed below:     |  |  | | --- | --- | | Sample use of Math Class Methods | In these methods of the Math class: The **Math.sqrt( )** method is used to calculate the square root of the parameter specified - num. The parameter passed may be an integer or a double, but if the answer is stored it must be stored as a double.  The **Math.round( )** method is used to calculate the rounded off value of the parameter specified - s. The parameter passed and the resulting value may be a double or an integer (although an integer should be used). |   The Math class has numerous methods that can be used for calculations in your program. See [the table](javascript:na_open_window('win',%20'math.htm',%2010,%2010,%20650,%20450,%200,%201,%200,%201,%201)) for a partial list and visit [Sun Microsystems](javascript:na_open_window('win',%20'http://java.sun.com/j2se/1.3/docs/api/java/lang/Math.html',%200,%200,%20700,%20450,%200,%201,%200,%201,%201)) for a complete list of Math methods.  **String Function Methods (Sun Microsystem String Class - found in the java.lang which is loaded automatically with every Java program)**  There are a number of methods that can be used to manipulate Strings as part of the *String* class. Interesting methods include the **length( )** and the **charAt( )** methods displayed below:     |  |  | | --- | --- | | Sample use of String Class Methods | In these methods of the String class (note the String variable itself is the object not the word String): The **str.length( )** method is used to calculate the length in characters of the string specified.  The **str.charAt(integer index position starting at 0)** method is used to return the character at the specified index position. The first character of a string is located at index 0, the next at index 1, with the last at index location length -1 (since we started counting at 0). |   The String class has numerous methods that can be used for string manipulation in your program. See [the table](javascript:na_open_window('win',%20'string.htm',%2010,%2010,%20650,%20450,%200,%201,%200,%201,%201)) for a partial list or visit [Sun Microsystems](javascript:na_open_window('win',%20'http://java.sun.com/j2se/1.3/docs/api/java/lang/String.html',%200,%200,%20700,%20450,%200,%201,%200,%201,%201)) for a complete list of String methods. |

|  |
| --- |
| http://angel.peelschools.org/AngelUploads/Content/1-Simler-2672-ICS3U0-A-Fall2010-BEL/_assoc/images/activities_heading.gif |
| **Assignment 8 - Using Predefined Methods** http://angel.peelschools.org/AngelUploads/Content/1-Simler-2672-ICS3U0-A-Fall2010-BEL/_assoc/A241925BEB714CC78951C5E8238B0C7B/images/line-mouse.gif  **(Evaluation Categories: Applications, Thinking/Inquiry, Communications)**   |  |  |  |  | | --- | --- | --- | --- | | **Be sure to properly indent and internally document each program (include comments). For questions 2-6 hand in rough work (IPO Chart, Pseudo Code and Java on paper). Hand in program listing, and program output for all programs.** | | | | |  |  |  |  | | http://angel.peelschools.org/AngelUploads/Content/1-Simler-2672-ICS3U0-A-Fall2010-BEL/_assoc/A241925BEB714CC78951C5E8238B0C7B/images/bullet-pencil.gif | **Steps** | | | |  |  |  |  | |  | **1.** | Using the Console class of graphics, draw a picture of a streetscape similar to what you would find in the downtown of your nearest town. Include buildings, sidewalks, store fronts, and whatever other detail you wish to include. | | |  |  |  |  | |  | **2.** | Write a program that asks the user to enter two real numbers. It should display the number on the screen and then proceed to calculate and displaying the following: The square root of the first number, the square root of the second number, the highest of the two numbers, the lowest of the two numbers, the two numbers multiplied together with the result rounded off, the first number to the second number power. | | |  |  |  | | |  | **3.** | Ask the user to input two strings. Display the strings in alphabetical order. Calculate and display the number of characters in the two words. Next convert the first string to all uppercase and the second string to all lowercase. Display the resulting strings. BONUS: Output the first string in reverse order (ie. Lucy becomes ycuL). | | |  |  |  | | |  | **4.** | Ask the user to input a sentence. Count the number of spaces in the sentence to determine the number of words entered. Output the number of characters in the sentence, the number of words in the sentence and the average number of characters per word. | | |  |  |  | | |  | **5.** | Write a program that asks the user to enter an angle (an integer between 1 and 360). Convert the angle in degrees to radians. (radians = (pi / 180) \* degrees) Calculate and display the cosine, sine and tangent of the angle. Have this program work for five different angles. BONUS: output the results in degrees instead of radians. | | |  |  |  | | |  | **6.** | Write a program that simulates the tossing of two dice. Toss the die, display what each dice displays and display the total of the two die. Have this program continue until a 7 or 11 is the total roll. BONUS: Count the number of rolls it takes to get a 7 or 11 and display the number of rolls it took. | | |

http://baigiang.violet.vn/present/show/entry\_id/4314295