**Day 1**

1. Check you have Java installed on your machine.
2. Install Eclipse.
3. Create a new project and write a "Hello World!" program. Make sure it runs :)
4. Write a program that prints the current date.
5. Write a program that prints the contents of a variable and the variable name for the following data types: int, float, double, boolean, String. Example output: age=35
6. Write programs for the following operations for 3 given integer: max, min.

**Day 2**

1. Write a program that prints the result of the following operations for 2 given integer variables: addition, multiplication, division, modulo. Try the same for 2 float variables and compare the results.
2. 1073741824 is a power of 2. Write program that prints what is the exponent.
3. Write a program which prints "Odd Number" if the int variable “number” is odd, or “Even Number” otherwise
4. Write a program which prints "ONE", "TWO",... , "NINE", "OTHER" if the int variable "number" is 1, 2,... , 9, or other, respectively. Use (a) a "nested-if" statement; (b) a "switch-case" statement.
5. Write a program that computes the factorial of 10. Use all 3 types of loops.
6. Write a program that prints the first number between 123456 and 125000 that divides by 27.
7. Write a program that print how many numbers between 123456 and 125000 can be divided by 27.
8. Sort the contents of an integer array.
9. Given an array of integers find the largest odd number in the array.
10. Print the command line arguments received by your program.
11. Create an array of Strings containing 10 names. Print the names with their index in the array. Example 1) John Smith 2)Bob Brown etc.

**Day 3**

1. Write a method that returns the longest String from an array of strings sent as parameter.
2. Write a method that returns the first String from a given array of strings, that begins with a given string. Example: ["The", "best", "party", "ever"] is the array and "par" is the given string. The method should return "party".
3. Write a method that transforms the string "we#make#great#code#" into "We Make GREAT code".
4. Define a class Person with the following attributes : first name, last name, age, address, phone number and email. Write a program that creates an array of persons and displays all the data for the persons in the array.
5. Define a class car with the following attributes: model, brand, year, number of km and driver. The driver should be of type Person. Create a program that displays the data of a car in the following format:   
   2006 Toyota Avensis,  
   230000 km,  
   John Smith, 35 years old  
   34rd Pasadena Avenue  
   phone: +1303577782  
   email: [jsmith@email.com](mailto:jsmith@email.com)
6. We consider 2 instances of the Person class to be equal if they have the same email address. Write a method in the Car class with the following prototype : boolean hasSameDriver(Car car);
7. Add 2 new methods to the Person class: is Older and isNeighbor (2 Persons are neighbors if they live on the same street). Create an array with instances of the Person class and try to find for the first person all his oldest neighbours.

**Day 4**

Today you will start your first Java project. You will need to apply the knowledge you gained the previous days. You will work on the project you start today in the following week and with each day we will introduce new concepts and techniques. So try to give your best and make an awesome project from the beginning :) .

**Geometrixx**

Geometrixx is a company specialized in selling geometric shapes. They want to enter into the digital era and start selling their shapes using a specialized software. They want you to help them create an inventory with their existing shapes. The first step will be to create the items they are selling and display the items in the console with all the informations :lenght, widht, radius, color, area, perimeter, price and quality.   
  
Two shapes of the same type can merge into one shape of the same type with increased quality.  
The shapes need to support basic geometric operations : translation and rotation. A Geometrixx can change it's color, but only with a color from an approved list: pink, purple, blue and grey

For start they need the following shapes:  
1. Circle: A plain old circle. Make sure it is round.  
2. Square: A plain old square.  
3. SuperCircle: It's a circle, but it is SUPER. Of course more expensive than a simple Circle and the area is calculated differently.  
4. OddTriangle: Works like a classic triangle, but when they merge the quality is rounded to the nearest odd number of the highest quality. So merging t1 with quality 1 with t2 with quality 3 gives a quality of 5.

Start working! The guys at Geometrixx want to start testing their digital shapes.

**Day 5 ##**

Now that you have learned about inheritance, try to remodel your classes so that you don't have duplicated code.

Use abstract classes and also interfaces. Now that you've finished with the rework, Geometrixx called and aksed to add a new property for all the shapes: creationDate. This information should also appear when you print a shape to the console. Make sure the creationDate is updated when you merge 2 objects.

**Day 6**

Now that you have some working shapes, the guys at Geometrixx want to have an inventory with all the available shapes. They need to be able to add new shapes in the invetory, delete from the inventory, update an item from the invetory and search by color, price, quality and type. Put those collections at work.

**Day 7**

Geometrixx is impressed with the work you delivered! They have although a problem: they have a newbie in the team and he doesn't understand how the inventory works. He messed up the inventory by introducing wrong data in for the objects: OddTriangle with an even quality, Supercircles at the same price as Circles, Squares without the lenght of the side etc.   
Help them by preventing this type of situation to occur by putting some exceptions in there. Try to treat as many situations as possible. Use your own exceptions.

**Day 8**

There is a new manager at Geometrixx and he is all about files. Everything is in a file from his point of view. So he wants the invetory to be loaded from a .txt file. Each line of the file will describe a shape. The delimiter for the attributes is "#". When the inventory is loaded, the user should be notified by a message stating the total number of shapes loaded and how many of each type.

Also when someone searches in the invetory the search result should be outputed to a .txt file. The output file will have a name matching this pattern:   
geometrix\_search\_DATE\_TIME\_NUMBEROFRESULTS.txt

Make sure the manager is happy so you get more projects!

**Day 9**

Oh, yeah! This new manager just called. He is impressed by your work so he wants to add new features to the application.  
1. The application will need to read from 3 different inventory files, one by one, and at the end display the total number of lines read from all the files toghether. 2. A new property needs to be introduced for the shapes: category.There can be only 3 categories: Basic, Extended and Composed. 3. Create a Map with the category as key and as value a list with all the shapes in that category. Provide a method that displays all the shapes in a specific category.

**Day 10**

You are close to finish this project! Geometrixx is growing into a corporation. So they need an enterprise application. Change the existing invetory application to store the inventory in a database and search in the database with SQL when someone wants to find a shape. Do not load all the shapes in the application memory!

Steps to start: Install a MySQL database server and MySQL Workbench. Download the JDBC MySQL Connector and add it to your projects classpath.