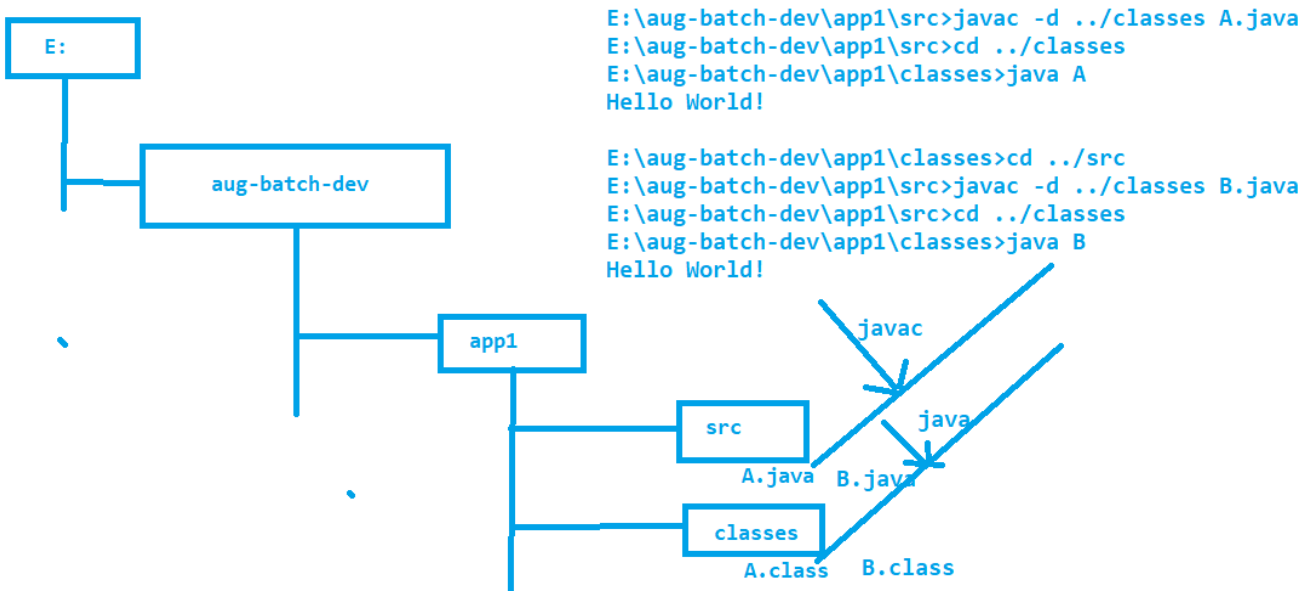
**Concepts: JDK installation, path, -d, -cp, println, print, keywords, identifiers, literals, bult-in and localvariables**

**Theory points**

1. Better to donload compressed format of JDK software.
2. After downloading compressed one, just extract into a folder by using winrar software.
3. All comands available in the bin folder of JDK.
4. Path is an environment variable. Its used for specifying commands locations to the operating syetem (command prompt)
5. Path can be updated in 3 ways
    1. We can set a path in the command prompt through set command. Its visible to current command window.
    2. We can set a path in my pc for a current user.
    3. We can also set a path to all the users of a current system in the my pc.
6. To read an environment variable from a command prompt, use the following command
    1. In windows => echo %env_variable%
       Like echo %path%

2. In linux => echo $env_variable
   Like echo $PATH
7. -d option is used to specify the different directory for the compiled class files. Its used with javac command
8. -cp option is used with a java command. Its used to specify the class files locations to the java command
9. -help option is used to know all other options and their usage description
10. -version is used to know the version of javac and java commands

11. Development structure




12. Programming elements

1. key words : the word which is already assigned with some meaning in the programming language.

   Keywords from simple program:

   class, public, static, void

   Note: All keywords are in the lowercase only.

2. literal : value

   int ==> 10, 200, 0, -67, 4545

   double ==> 1.456, 9.0, 5.6, -100.89, 4.5, -0.9, 5.2

   char ==> 't', 'W', 'p', '7', '#', '$', '5'

   boolean ==> true, false

   String ==> "abc", "hello", "India is great", "123", "#$%^Hello", "", " ", "Hello456"

3. identifiers: names

   1. first char : any alphabet (or) _ (or) $

   2. second char onwards : any alphabet (or) _ (or) $ (or) any digit

   3. no keyword can be an identifier

   **valid identifiers:**

   Hello, _Hello, H_llo, $Hello, Hello1, H5llo, Hell_o, H$ello, hello, helLO

   **In-valid identifiers**

1Hello, Hello All, Hello&Hi, hell0%hi, void, class, public

**Observations:**

1. Digits allowed only from 2nd char onwards.
2. Only 2 spl chars are allowed. Those are
   _ and $
3. Space is not allowed in the identifier
4. built-in : pre defined features

System.out.println

13. System.out.println() is having dual features. First one is printing specified content and second one is producing new line
14. We can use no argument System.out.println(). Here it produces a blank line.
15. System.out.print() is having only one feature. It's a printing feature. It will be retaining in the same line after printing. It won't go to the next line.
16. We can't use no argument System.out.print()
17. Every variable should have a datatype and the identifier(name)

18. Variable is a container where we can place one value. This value can vary in the different statements. But at any statement, there will be only one value.
19. Variable value can reassign any number of times
20. Same type of multiple variables can be declared in the same statement by using comma as a delimiter. Like
    int a1 = 10, a2, a3 = 20;
21. Same type of multiple variables can be assigned with the same value in the same statement. Lie
    int b1, b2, b3 = 10, b4;
    b1 = b2 = b3 = b4 = 200;
22. Variable can be declared and assigned in different statements. Like
    int y1;
    y1 = 100;
23. A variable which is declared inside a method is called a local variable.
24. There are some conditions to be followed with respect to local variables.
    1. Local variables are not initialized with any default values. Local variables should be initialized before usage. If you use local variable without initialization,

then you will get a compilation error. Compilation error message: Variable might have not initialized

2. Variable type and assigning value type should be same. Otherwise you get a compilation error. Compilation error message: incompatible assignment
3. Forward reference is not allowed. The following code gives compilation error

   System.out.println(i);

   int i = 10;

   Compilation error Message: can't find symbol i.
4. Two variables with a same name in the same method is not allowed. Its not allowed even data type is changed and values are changed. Every variable should have different names in the same method. If two variables are same in the same method, then compilation error. Compilation error message : variable already declared with the same name

## FAQS (Frequently asked questions):

1. What is path?
2. Why path is required?
3. How many ways are there to update a path?
4. Which way of updating a path is most preferable?
5. Which command is used to update a path in the command prompt?
6. Which command is used to read the environment variable in the command prompt?
7. How to separate class files from java files while compiling?
8. Why do we required –d option?
9. How to specify class files location to the java command?
10. Why do we require –cp option?
11. What is the option used to know all available options of javac command?
12. Which option is used to know the version of javac command?
13. What is a keyword?
14. Tell me the keywords from Helloworld program?

15. What are the boolean literals?
16. Whether single digit can be enclosed in a single quote to make a char type or not?
17. What could be the data type of "12345" ?
18. Is it a valid literal, if we keep like '12' ?
19. What is the difference between 10 and 10.0?
20. Is it possible to develop a string without any characters?
21. What is an identifier?
22. Which spl chars allowed in the identifier?
23. Where digits are allowed in the identifier?
24. Keyword can be an identifier?
25. Whether space is allowed in the identifier?
26. What are the differences between println and print from System.out?
27. What is a variable?
28. Why do we require a variable?
29. What are conditions to be followed with respect to local variable?
30. What could be data types to a local variable which is used to store "hello"?