1. Why java is popular

* Robust
* Platform independent
* Lightweight
* No pointers => addresses
* OOPs concepts / principles
* Multithreaded
* Distributed environment
* Scalable
* Secured

1. Compilers and interpreters?

Compiler : check the complete code for syntax errors and if none then converts to some intermediate format

Hello.java => compiler => Hello.class (bytecode)

Interpreter -> read the code(.class) line by line and execute them to machine executable

1. JDK -> java dev kit [ comes up with the java libraries and all the excutable ]

JRE -> java runtime env [ provides with environment for java applications to execute and manage all java applications running]

JVM -> java virtual machine [ 1 instance of jvm per application ]

1. Packages : group of similar kind of classes

* Package names should all be small. And it is the 1st statement in the file except comments
* Class names every word should always start with uppercase

1. Diff between print and println
2. Data types - 8
   1. Byte
   2. Char
   3. Short
   4. Int
   5. Long
   6. Float
   7. Double
   8. Boolean – true or false
   9. String – derived data type
3. Variables
   1. Naming conventions
   2. Declaration and initialization
   3. For char wrap the values around ‘’
   4. For String wrap the values around “”
4. Operators
   1. Unary ++ -- - !
   2. Binary
      1. Arithmetic + - \* / %
      2. Relational < > <= >= == !=
      3. Logical & && | || So && and || are called short circuit operators
   3. Ternary ?:
5. Escape sequence : \n \t \” \’ \\
6. Conditional constructs
   1. If-else
   2. Switch-case
7. Looping constructs
   1. For
   2. While
   3. Do-while
   4. Break
   5. Continue
   6. Nested loops
8. Arrays
   1. collection of data of same type
   2. it is stored in continuous memory locations
   3. size once declared is fixed
   4. uses [] syntax to create arrays
   5. arrays support indexing that starts from index 0 to length-1
   6. Arrays have a length property that displays the total size of the array
   7. Depending on what type of array it is, it has a default value
      1. Primitives : primitive defaults ex: int 0, double 0.0
      2. Reference types like String or user defined => null
9. For Each loop
10. String
    1. == and equals
    2. In built methods
11. Class
    1. User defined data type
    2. Blueprint
    3. Composite data type
    4. Class can have data members and member method
12. Methods
    1. Methods can take input in the form of parameters in the ()
    2. Methods that do not return anything must be declared as void
    3. Methods give output by “return” keyword
    4. There can be more than 1 conditional return statements and only 1 of them is executed at any given point of time
13. Constructors
    1. is a special method with the name same as the class name
    2. they DO NOT have return types not even void
    3. They are automatically invoked when the object of the class is created using new
    4. They cannot be invoked using .
    5. They are invoked only the 1st time the object is created
    6. purpose is to initialize the data members of the class
14. Getters and Setters
    1. Getters are accessors with return type and no parameters  
       public int getEid(){return eid;}  
       For bolean properties method starts with is and not get
    2. Setters are modifiers with void return type and takes 1 parameter  
       public void setEid(int eid){this.eid = eid;}
15. toString()
    1. provides with the string representation of the object
16. Constructor Overloading : static polymorphism   
     More than 1 method with the same name, but with
    1. Different number of arguments
    2. Different types
    3. Different sequence
    4. Changing the return type or access specifier or method name is NOT OVERLOADING
    5. Can be done in same class or parent-child as well