**Bits & Bytes of JAVA:**

**DAY 0(SEP 14): Induction Program**

**DAY 1(SEP 23): JAVA Fundamentals**

**Java: High level, Object oriented programming language, Platform independent, Strongly typed.**

**Java is more secure**

**Can be used for developing standalone apps, wed applications, android apps and enterprise applications**

**Byte code can run on any operating system post compilation.**

**Object: Instance of a class**

**In order to use any action or field in a file we need to access through object.**

**Class: It is a blueprint that object needs to follow or It is a combination of methods and attributes.**

**Example;**

**Class employee**

**{**

**ID,first name, last name, Phone(These are the attributes of a class)**

**{**

**And a class can also contain methods**

**How can a object be created?**

**It is created using a constructor(Constructors creates instance of a class that is object)**

**It is for creating and initializing the object.**

**\*\*\*In Java, always class name and file name should be same.**

**By default constructor takes zero argument as parameter in not explicitly defined.**

**JVM: Java virtual machine takes zero argument by default and provides it to file.**

**JVC: Java compiler**

**Data types:**

**Char – 2 bytes**

**Byte – 1 byte**

**Short – 2 bytes**

**Int – 4 bytes**

**Float – 4 bytes**

**Long – 8 bytes**

**Double – 8 bytes**

**Boolean – 1 bit**

**What is an abstract method?**

**It is a method that is only declared but not defined. It is the responsibility of child/sub class to define it. Or child/sub shall pass this method to next sub levels where it needs to defined by those methods**

**Abstract is a non-access modifier.**

**Parameter & Argument:**

**Parameters are at the receiver end while inputting.**

**Arguments are the values which are passed to the functions/methods.**

**Operators:**

**Arithimetic, logical, Increment/decrement, Assignment, Comparision, bitwise**

**OOPS:**

**Data abstraction**

**Data encapsulation (Security / Data hiding)**

**Polymorphism(Method overloading)**

**Inheritance(Re-usability)(Method overriding)**

**Conditionals:**

**If**

**If-else**

**Switch**

**We use switch when we have multiple if-else conditions to execute.**

**Switch(Condition){**

**Case 1:**

**Break;**

**Case 2:**

**Break;**

**Case 3:**

**Break;**

**Default:**

**}**

**Default statement is not mandatory in switch condition.**

**Default statement can be situated anywhere in the switch. It is allowed.**

**The value of a case should be constant. Only primitive types we can use switch like usual data types, enums and strings.**

**A switch can take an expression**

**Cases will take int value, characters and string type as well**

**String is a class type in java**

**Non-primitive types are user defined data types**

**Loopings: for, while, do-while**

**For(), while() and do while():**

**Iterators**

**Example:**

**1)**

**For(i=0;i<5;i++){**

**Sop(a[i]);**

**}**

**2)**

**I=0**

**While(i<5){**

**I++;**

**}**

**3)**

**Do{**

**I++;**

**}while(i<5);**

**//Do-while will execute one more time than while**

**For(int=0;i<5;i++){**

**Sop(i)**

**}**

**Int i=0;**

**While(i<5){**

**Sop(i)**

**I++;**

**}**

**Int i=0;**

**Do{**

**Sop(i);**

**I++;**

**Sop(i)**

**}while(i<5);**

**In a java program objects communicate by invoking methods.**

**Each object has a state/variable/attribute/field and behaviour/method/function/action.**

**Example:**

**Here the object is dog,**

**State: name, breed, size, weight**

**Behaviour: bark, jump, run, sit**

**DAY 2(SEP 24): JAVA Fundamentals**

**Comments:  
// ---------------------------------------// -> Single line comments**

**/\*---------------------------------------\*/ -> Multiline comments**

**While using a class first letter of each word should be capital. Ex: MyClassNum**

**While using a method first letter should be smallcase and from second word first letter should be capital. Ex:collegeHolidayStudents()**

**Method always has parenthesis**

**In java both class names and file names are case sensitive.**

**Public static void main(String[] args)**

**Public: Available both inside and outside package**

**Static: Method that can run without creating an object or which can be globally used**

**Void: return nothing**

**Main: Place where programs start execution**

**Parameters are declared inside parenthesis**

**Do not use ; after classes and methods**

**We use comments to increase the code readability and to avoid confusion while solving complex programs.**

**All the names that we use in java are known as identifiers.**

**Char – ‘a’ -> It is defied in single quotes**

**String – “str” -> Is is defined in double quotes**

**Scanner object is used for inputting data from user in java.**

**In a file one class must be public.**

**Constructors: By default zero argument constructor will be passed by jvm if not defined explicitly.**

**Casting:**

**Array: 1-D**

**DAY 3(SEP 27):**

**\*this keyword**

**\*super keyword**

**\*static keyword**

**\*method overloading & method overriding**

**static variable can be accessed globally(Globally).**

**It is not a instance variable. Instance means not a part of an object. It is a class variable which can be accessed globally.**

**We can have variables, classes, blocks{….},methods(Main() method is static) 🡪Can be static**

**To count number of objects we can use static to save memory.**

**this operator: It is a reference variable which points to the current object.**

**It can used on instance variables, passed as an argument, with constructor also**

**Super: It is a reference variable which can point the super class/parent class objects.**

**It can be used with methods, constructors and variables.**

**Package is a place where all our programs are available. We can import, 2 packages can have same name**

**Count hold common value across the objects**

**we can use a static variable in a non-static method.**

**Any static method is not referenced used an object**

**To call a non static method we have call it through an object. If everything is static in a program then no issues.**

**Methods are defined using parenthesis**

**If your method is of int type then you can have integer value or else the compiler complains**

**There is only one memory allocation done for the static variable, there is only one reference possible for the static variable**

**Method overloading bothers on number of parameters and type of parameters.**

**Method overriding needs inheritance**

**Access modifiers:**

**1)Public**

**2)Private**

**3)Protected**

**4)Default**

**DAY 4(SEP 28):**

**Access modifiers: To define the visibility of code**

**Abstraction concept:**

**Abstract class VS Interface**

**Extends VS implements**

**There are 4 types of access modifiers**

**1)public**

**It can accessed all over the package and outside the package as well**

**No big restrictions**

**2)Private: Only defined to use in a particular class.**

**This is not available for the other class in the same file**

**Heavily restricted**

**3)Protected**

**This is available for sub classes but not outside file.**

**This little restrictive**

**4)Default**

**This is available all over the current package**

**Non restricting**

**Interfaces: There are 100% abstract class. It will have public static final and followed by variables, All the methods are abstract, only declarations made but not definations.**

**DAY 5(SEP 19):**

**Exception handling and file handling**

**There are different type exceptions in java:**

**In File handling we can perform reading and writing data in multiple ways:**

**1)**

**To try catch inside a catch**

**2)**

**One try multiple catch block**

**3)**

**differences between the several file classes like**

**BufferedReader**

**BufferedWriter**

**InputStreamReader**

**OutputStreamReader**

**InputStreamWriter**

**OutputStreamWriter**

**Filewriter**

**Filereader**

**FileOutputStream**

**FileInputStream**