|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Source Code vs Object Code:**  Source code is written by programmer and it is human readable . Object code is compiled file. It is created when source code is compiled. It is machine readable.  **Using Compiler with flags:**  Use the flag while testing the code during development process and check contents of variable.  **Keywords and Variables:**   * Keywords are set of reserved words in a programming language like if , else , switch , abstract , super.  |  | | --- | | * Variables are like containers which are used to store the data. Variables can have different datatypes like integer, String, float etc. | |  |   **Variable Scope, Lifetime Initial value and modifier:**   * Scope means accessibility of variable. Two types of scope-class level scope and method level scope. Variables are accessible within the scope. * The lifetime of a variable is the time during which variable stays in memory If we don’t initialize variable with some value, then default value is set to that variable.   **Programming Constructs(Selection constructs and Loop Constructs)**   * Loop constructs allows code to be executed zero time or many times. E.g. for loop, while loop, do-while loop. * Selection construct execute the code with certain condition. E.g. if-else   **Array:**  Array is collection of similar data type. It can be of any data type like string, integer, etc. In java array can be defined as:  int a[] = new int[(size of an array)]  **Store data in array and process elements:**  Array is indexed-based and we can store only a fixed set of data in array.  **Pointers, Declare and initialize pointers:**  Pointers is address of some location in a memory. In java pointers are also called as reference variables and It can not be controlled by developer. But on risk we can use “unsafe” method to control pointers.  **Functions definitions/Invocations:**  Functions are a block of code which performs a specific task and can be invoked from anywhere within the scope.  **Call Functions by passing value and passing reference**   * Call by Value means calling a method with a parameter as value. Through this, the argument value is passed to the parameter.  The values of the arguments remain the same even after the method invocation. * Java uses only call by value while passing reference variables as well. It creates a copy of references and passes them as valuable to the methods and changes are reflected outside the method as well.   **Dynamic memory allocation/Deallocation:**  When memory is allocated at runtime then it is called as dynamic memory allocation.  Memory Deallocation is done by garbage collector.Memory deallocation is done when object has no longer any refernces. Jvisualvm and jconsole tool are used for memory analysis.  **Memory Leakage:**  Memory leakageaccurs when **objects are no longer being used by the application, but the Garbage Collector is unable to remove them from working memory.**  **User Defined Structures:**  **User defined structure is the collection of different datatype. Class, structure and union are the examples of user defined structures.**    **Access data members:**  **User Defined data structures contains data members and member functions.** These data members can be accessed by the member functions and getters and setters.  **Dynamic Memory Allocation for user defined structures**  Memory can be allocated to the objects using the constructors and default values or specified values are initialized for the data members.  **Array of Pointers**  In java array of object is used to store multiple object of the class.  **Iterate over array of Pointers**   |  | | --- | |  | | Array of objects can be iterated using the for loop and objects can be printed in a specific format by overriding the toString method. | | |  |   **Implement Data Structures**  Implementation of data structures like array , linkedlist , stack , queue , binary tree.  **LinkedList**  Linkedlist is the linear data structure having sequential access to elements , dynamic memory allocation and elements can be stored and retrieved from anywhere in the linkedlist.  **Stack:**  Stack is a linear data structure having sequential access to elements , dynamic memory allocation or static allocation depends on the type of implementation and elements can be stored and retrieved only at the top of the stack.  **Queue:**  Queue is a linear data structure having sequential access to elements , dynamic memory allocation or static allocation depends on the type of implementation and elements can be stored and retrieved only at the front and rear of the queue.  **Binary Tree:**   * Binary trees are hierarchical data structures in the form of roots and its children where each node can have at most two children. * These have sequential access and dynamic memory allocation. There are two types of traversals BFS and DFS.  |  | | --- | |  | |  |  | |  | |  | |  | |  | |  |  | |  | |  | |  | |  | |  | |