int binarySearch(int arr[], int l, int r, int x) {

// base case: empty list

// r: last; l: first

if (r >= l) {

int mid = (l+r)/2;

// base case

if(arr[mid] == x) // ask you to fill out code here

return mid;

if(arr[mid] > x) // on the left section

return binarySearch(arr, l, mid-1, x);

else // on the right section

return binarySearch(arr, mid+1, r, x);

return -1; // element not found

----------------------------------------------------------------------------------------------------------------------------------

final static int ***LIMIT*** = 10;

Compute Sum Code

public static int compute(int anArray[]) {

int sum\_array = 0;

if (anArray.length < ***LIMIT***)

return 0;

int i = 0;

while (i < ***LIMIT***) {

sum\_array += anArray[i];

i++

return sum\_array;

-----------------------------------------------------------------------------------------------------------------------------------public class MyThread extends Thread

public void run() {

Calendar cal = Calendar.*getInstance*();

SimpleDateFormat sdf = new SimpleDateFormat("HH:mm:ss");

System.***out***.println("MyThread.run(): " + sdf.format(cal.getTime()))

* Recursion breaks a problem into smaller identical problems
* Each recursive call solves an identical but smaller problem
* A static method is a class method
* Non-text files are called binary or general files
* Text file is a file of characters organized into lines
* Invariant -> A condition that is always true at a particular point
* Loop Invariant -> A condition which is true before and after the execution of each loop
* A solution is good if the total cost incurred over all phases of it’s life cycle is minimal
* Encapsulation -> Objects combine data and operations
* Inheritance -> Classes can inherit properties from other classes
* Polymorphism -> Objects can determine appropriate operations at execution time
* Abstract Data Type(ADT) -> A collection of data and a set of operations on the data
* An ADT’s operations can be used without knowing how the operations are implemented. However, operation specifications should be known.
* Within problem solving, ADT’s support algorithms and algorithms are part of what constitute ADT’s
* All modules and ADT’s should hide something
* Encapsulation hides inner details
* Unified Modeling Language (UML) is a modeling language used to express object-oriented designs
* Modularity isolates errors
* Constructor -> Is a special kind of a method which has the same name as the class and no return type. Is executed only when an object is crated.
* For and While loops are equivalent. However, For is much more used and preferred than the While loop.
* Files provide both sequential and random access; Arrays only provide random access
* Files grow; Arrays remain constant with regards to size.
* Selection Statements -> if, switch; Iteration Statements -> while, for, and do-while
* String Tokenizer -> Strings that can be broken into pieces called tokens
* Object Oriented Design (OOD) -> Produces modular solutions for problems that primarily involve data. Identifies objects by focusing on the nouns in the problem statement
* Functional Decomposition (FD) -> Produces modular solutions for the problems in which the emphasis is on the algorithms. Identifies actions by focusing on the verbs in the problem statement. In FD, a task is addressed successively at lower levels of detail.
* Use OOD and FD in tandem to formulate modular solutions.
* Procedural Abstraction -> Specifies what to do, not how to do it. Separates the purpose of a method from it’s implementation. Each method will be/should be specified before it is implemented.
* Data Abstraction -> Specify what you will do to data, and not how to do it.
* Implicit Conversions/Integral Promotions -> If the operands of an operator differ in data types, the data type that is lower in the hierarchy is converted to one that is higher. Example: int -> long -> float -> double
* Modules -> Self contained units of code. Are designed to be loosely connected and highly cohesive.
* Methods should check their invariants and enforce their pre-conditions.
* Object-Oriented Programming (OOP) allows re-use of existing classes, thus saving development time.
* Specifications do not indicate how to implement a method.



Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated



