|  |  |
| --- | --- |
| COURSE | BDSE |
| QUALIFICATION NAME | BDSE |
| MODULE NAME | SOFTWARE ENGINEERING |
| ASSIGNMENT | ASSIGNMENT 2 |
| ISSUE DATE | 24 DESEMBER 2022 |
| SUBMISSION DATE | 12 JANUARY 2023 |

|  |
| --- |
|  |

Table of Contents

TASK 11-3

TASK 24-8

TASK 39-13

REFERENCE14

Task 1.

1. Document the architecture of the system along with modul and its component description.

To begin with, we need to comprehend the issue that the program is supposed to answer. Additionally, any guidelines that must be followed. In this instance, North Sussex Judo has been asked to use a program that must estimate the monthly cost of training fees for its athletes. This program was created by entering data, which is made up of figures in each training session, including the session fee, and will be used as information to calculate the overall monthly cost.

Based on the data, we plan to suggest the following system architecture for the project:

The program will ask the user for the necessary data, such as session data from each training session, and will utilize that information to determine the overall cost for each month before showing the user the final cost.

Modules: The software will be broken up into several modules, such as classes, methods, variables, etc.

classes: This program has a class for athletes, which holds information about each athlete, including name, goals, current weight, competition weight category, and number of competitions.

Method:

\*Accepted user inputs: Make it simple to be in charge of accepting user input From the user, such as number of training sessions and the cost/sessions.

\*The overall cost for the month will be determined by thecostcall based on user input

and

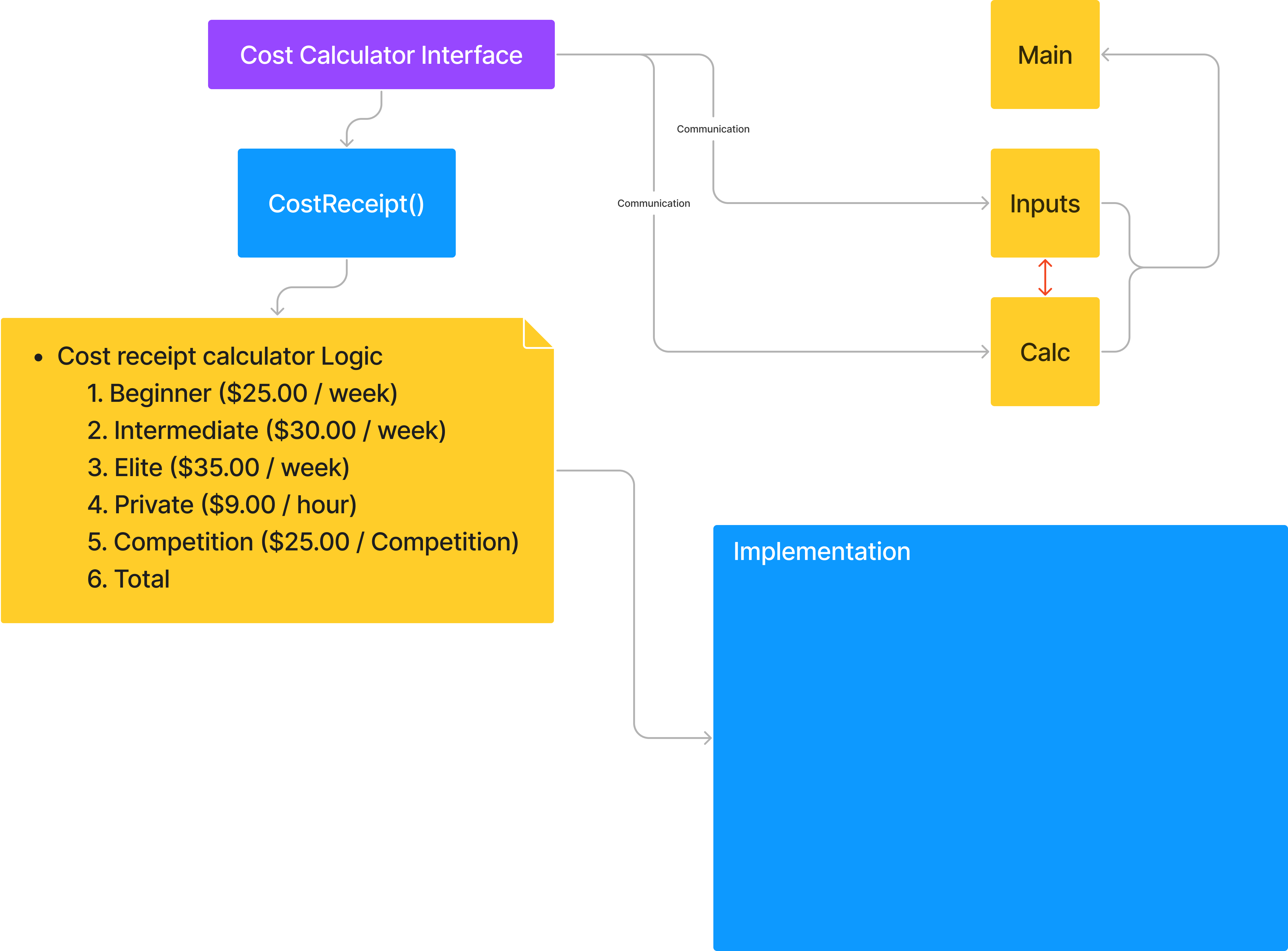
\*the output display will be computing cost to the user, along with athlete’name and an itemized list off all cost for the month.

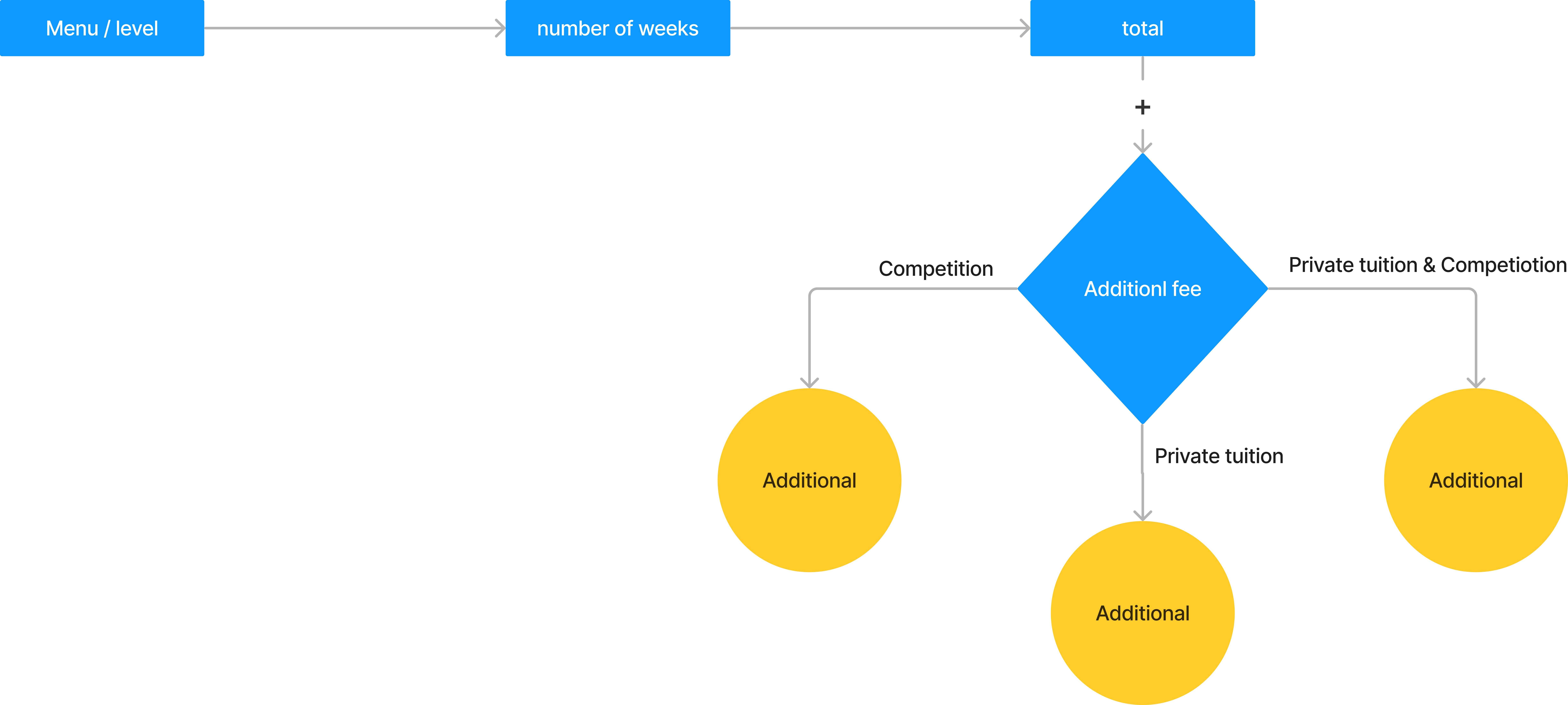
Variable: This software uses the variable to store data, including the quantity and cost of sessions.

Object: A produced object by the athlete class software to denote a certain athlete.

For instance, each of the items in the athlete class makes the class unique.

The NSJ training cost calculator will be a program that allows users to input data about the athletes and calculates the total cost of training fees for the each months. The program will consist an athlete class. Three main method will be accepting the input. Calculating the total cost and display the results. And variable to store data, it also create the object to represent the specific athletes and have a unique instance for each object.





. Task 2: Algorithm

* An algorithm is a collection of instructions that a computer must follow in order to carry out calculations or other problem-solving tasks.
* A formal definition of an algorithm states that it is a finite set of instructions that are followed in a particular order to complete a specified goal.
* It's not the whole program or code; rather, it's a flowchart or other informal depiction

of the problem's straightforward logic.

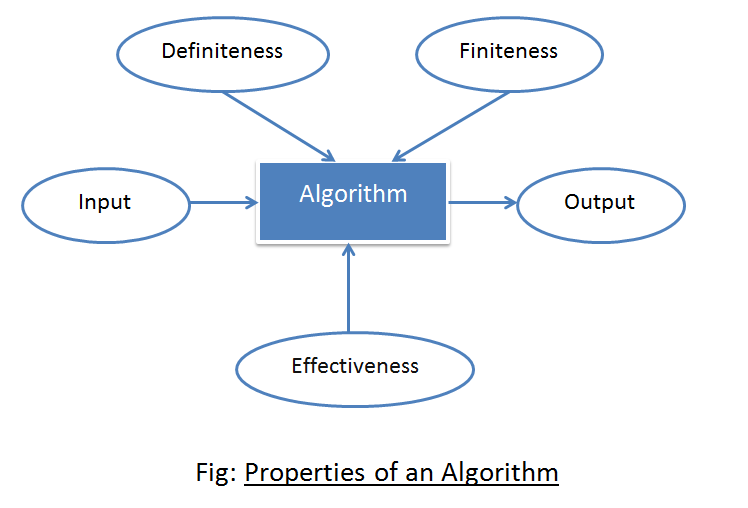
Each algorithm has the following characteristics:

* Input: Each algorithm needs some input data. A value other than 0 may be provided as input to an algorithm.
* Output: An algorithm will produce one or more results at its conclusion.

• Unambiguity: A perfect algorithm must be unambiguous, which implies that its directives must be plain and simple.

• Finiteness: A finite algorithm is required. In this context, "finiteness" refers to the requirement that an algorithm have a finite number of instructions, or instructions that can be counted.

• Effectiveness: An algorithm must be effective because every instruction has an impact on the whole procedure.



* Benefit algorithm :
  1. Better problem solving
  2. Improves efficiency
  3. Provides clarity
* Type of algorithm :

1. Brute Force Algorithm: This is the most fundamental and straightforward type of algorithm. A Brute Force Algorithm is the simplest solution to a problem, or the first solution that comes to mind when we see the problem.

2. Algorithm Recursive:

Recursion is used in this type of algorithm. In recursion, a problem is solved by breaking it down into subproblems of the same type and repeatedly calling own self until the problem is solved with the help of a base condition.

3. Randomized Algorithm: A random number is used in the randomized algorithm. It aids in determining the expected outcome.

4. Algorithm for Sorting:

Sorting algorithms are used to arrange data in ascending or descending order. It is also used to organize data in an efficient and useful way.

5. Searching Algorithm: The searching algorithm is the algorithm used to find a specific key in a set of sorted or unsorted data.

6. Hashing Algorithm: Hashing algorithms work similarly to Searching algorithms, but they include an index with a key ID, i.e. a key-value pair.

7. Divide and Conquer Algorithm: The algorithm divides the problem into subproblems, solves one subproblem and the final solution to get the final solution. It consists of three steps:

- Splitting

- Solving

- Combining

8. Greed Algorithm: In this algorithm, the solution of next part is made based on the immediate profit of the next part.

9 Dynamic programming algorithm: uses the concept of using already found solutions to avoid recalculating part of the problem. It breaks down a problem into smaller sub-problems that fit together and solves them.

10. Backtracking algorithm: builds a solution by searching all possible solutions. We will proceed with the solution according to the norms.

Conclusion: An algorithm is a problem-solving procedure language or formula. The issue is based on a survey.

A series of specified actions. A computer program can be viewed as a complex algorithm. In the fields of mathematics and computer science. An algorithm is typically defined as a short procedure that solves a recurring problem. Algorithms are widely used in all areas of information technology. As an example, a search engine algorithm accepts keyword and operator strings as input. It will also search its associated databases for relevant web pages and return the results.

Advantages of algorithm :

* easy to understand.
* Algorithm is a step wise representation solution to given a problem
* In algorithm the problem is broken down into smaller pieces of steps hence, it is easier for the programmer to convert it into an actual program

Disadvantages of algorithm :

* Writing an algoritm takes a long time.
* Understanding complex logic through algorithms can be difficult
* Branching and looping statements are difficult to show algorithm.

START

(MENU)

INPUT ATHLETE NAME

Display again

DISPLAY TRAINING PLAN

Beginner ($9.00) / WEEK

Intermediate($30.00)/WEEK

Elite ($35.00)/WEEK

NON VALID

enter

VALID

PLEASE INPUT YOUR WEIGHT

CHOSE WEIGHT COMPETITION CATEGORY

Heavy-weight (over 100 kg)

Light-weight (100 kg)

Middle weight (90 kg)

Light-middle weight(81 kg)

Light-weight(73 kg)

Flyweight(66 kg)

YES

valid

DISPLAY

DISPLAY AGAIN

Non valid

ADDITIONAL IF ELSE

INPUT

NON VALID

VALID

PLEASE INPUT HOUR YOU WANT

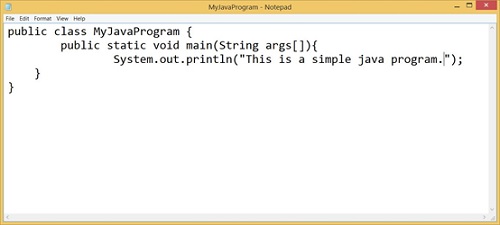
ACCUMULATE & CALCULATE TOTAL COSTING

AFTER COMPARE WEIGHT + WEIGHT COMPETITION CATEGORY

CALCULATE/MONTH

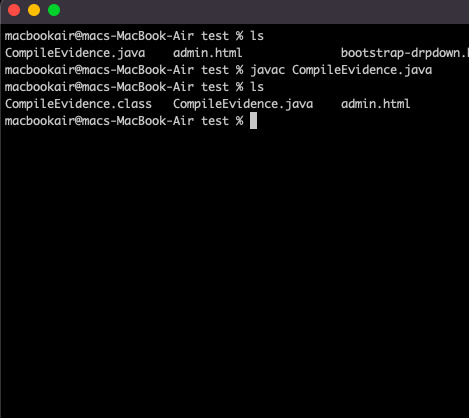
CALCULATE/MONTH

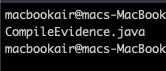
Task 3.



Step 1: Write a Java program:

By writing a Java program on any editor or IDE, we mean creating a Java program. This includes modifying the program on the editor after it has been written once. The creation of a Java program is not limited to simply writing the program on an editor and then leaving it. In general, creating a Java program entails writing a program in an editor or IDE, making all necessary corrections, and then saving the program to a secondary storage device such as a hard drive.





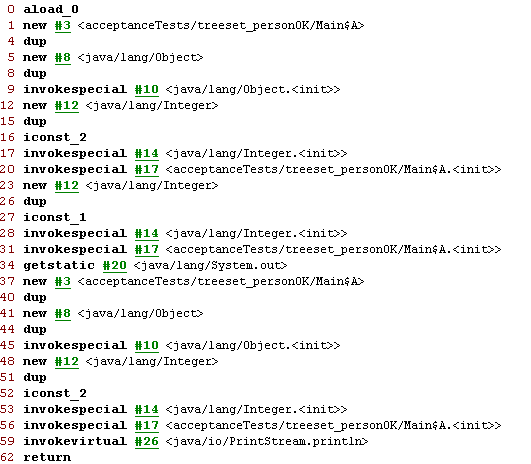
Step 2: Create a Java Program

Following the creation of the program, the compilation of the Java program is our next step. The compiler generally compiles the Java program that we created in step 1 with a.java extension. Assume we have a program called WelcomeJavaPrograms.java, and we want to compile it using a command like javac. It is known as the Java compiler. We compile a Java program with the.java extension after opening the command prompt or shell console.

javac WelcomeJavaPrograms.java.

If no errors occur in our program after running the javac command, we will receive a.class file containing the bytecode. This indicates that the compiler successfully converted the Java source code to bytecode. The JVM typically uses bytecode to run a specific application. JVM is a virtual machine that sits between our bytecode and the underlying operating system. When creating a Java program, you should provide the executable file that only the JVM understands. To file, add a java extension. It denotes that the file in question is Java source code.

## Bytecode



## JVM loads the program into memory:

## Step 3 :

## JVM needs memory to load the.class file before it can execute it. Loading refers to the process of storing a program in memory so that it can be executed. The JVM includes a class loader, whose primary function is to load the.class file into primary memory for execution. The class loader loads all of the.class files required by our program for execution into memory just before execution.

## Loader into memory & Execution Engine

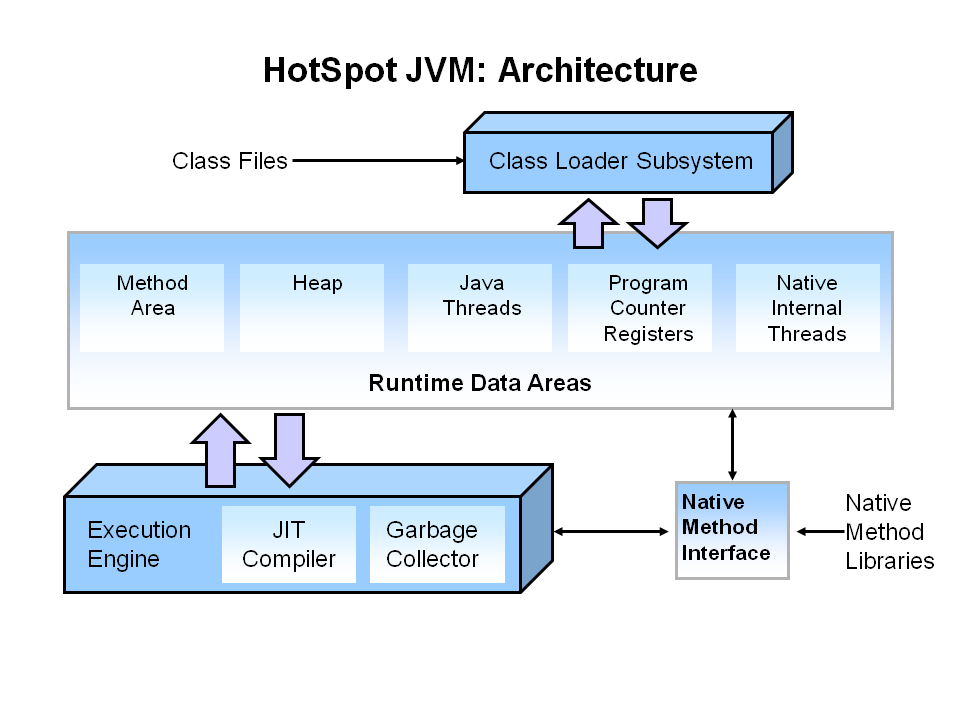
## Step 4: JVM Verification of Bytecode:

## JVM includes a bytecode verifier to ensure program security. After the classes are loaded into memory, the bytecode verifier enters the picture and verifies the bytecode of the loaded class to ensure security. It determines whether bytecodes are valid. As a result, it protects our computer from malicious viruses and worms.

## Step 5: Run the Java program:

## JVM executes the actions we've written in our Java program by interpreting bytecode. Old JVMs were slow and only executed and interpreted one bytecode at a time. Modern JVMs employ a JIT compilation unit, also known as just-in-time compilation. These compilers run multiple instructions in parallel. Because JVM uses them to find hot spots in bytecode, they are also known as Java HotSpot compilers. By hot spots, we mean that the JVM analyzes bytecode by looking for places in the program that are frequently executed. These are quite common.

## The Java HotSpot compiler converts executable bytecodes to faster machine language code. The significance of this is that whenever JVM encounters such parts in bytecode again, it processes machine language code, which is much faster than executing one byte code at a time.



CREATE THE PROJECT of

JVM

+

COMPILE : JAVA to class file

initialization

lingking

loading

MEMORY

EXECUTION ENGINE

## references :

## <https://docs.gradle.org/current/userguide/building_java_projects.html>

## <https://data-flair.training/blogs/java-virtual-machine-jvm/>

## <https://www.geeksforgeeks.org/introduction-to-algorithms/>