# Write up for Group C assignment

- Roll Numbers: **2459, 2465,2466**
- Name of Project: **Application to find jobs/internships as per preferences of applicant.**

➢ Problem Statement: To make an application to find jobs/internship where a recruiter can post his job advertisement and a job finder can easily find all the jobs which fits in search bracket of a particular individual and implement it with efficient search algorithm by using hash table and a graph with file use for storage management.

➢ Details about designing: Classes, Functions

Classes and their functions:

1. **Project**(): main class to call all the functions as per the recruiter or user needs.

   Functions:

   a) `public static void main(String args[]);`

   Main function to make function calls.

2. **Graph**(): The graph which defines basic route for filters of searching is constructed in this class. It gives value for hash function calculation.

   Functions:

   a) `hash accept();`

   Takes all the inputs from the recruiter and helps recruiter to post a/an job/internship.

   b) `hash search_accept();`

   Takes all the inputs from the job finder and helps job finder to find a/an job/internship.

   c) `static public hash display_hash(hash o);`

   Displays the preferences which the recruiter posts.

   d) `static public void display_search_hash(hash o);`

   Displays the preferences which the recruiter had posted when a/an job/internship is available for job finder.

   e) `String validation_graph(node o);`

   To validate the input given by user.

   f) `String validation_main(String s1,String s2);`

   To validate the input given by user.

   g) `public hash set(int, int, String, String, String, String, String, String, String, String, String, String);`

   Fills all the data for a hash object and its storage.

3. **Hash**(): The hash tables and its operations are performed in this class.

   Functions:

   a) `public hash()`

      Constructor which defines initial values for this data type.

   b) `void copy_previous_data();`

      Copies the previous data i.e. all the previous entries from the file into the hash tables.

   c) `void write_new_updates();`

      Update file with new entries for storage.

   d) `public void add(hash h);`

      To add entry in hash table.

   e) `public int calc_add(hash h);public int calc_add(hash h);`

      Takes all the fields selected by the recruiter/job finder and generates an address for the storage of entry by the recruiter.

   f) `public void search(hash h);`

      Search whether a particular entry is present in hash table.

   g) `void print_all_jobs();`

      Prints all the jobs posted.

4. **Re_node**(): Data type to store the details of job/internship.

   Functions:

   a) `void assign(String n,String m, int no,int add)`

      Assigns values to the entities of job/internship.

5. **Vector_obj**(): This class implements Serializable and is the data type to add objects in files which act as a database.

   Functions:

   a) `public vector_obj()`

      Constructor which defines initial values for this data type.

   b) `void initialize(int i)`

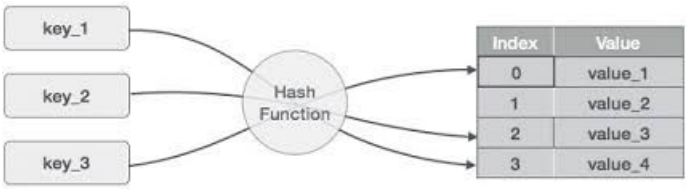
      Method to initialize number of objects.

   c) `void assign(int s,int d,int a, int list_no,String sec,String dom,String ji,String pl,String t,String e,String ys,String ex,String sal,String cn,String loc,String jd)`

      Assigns values to the entities of job/internship.

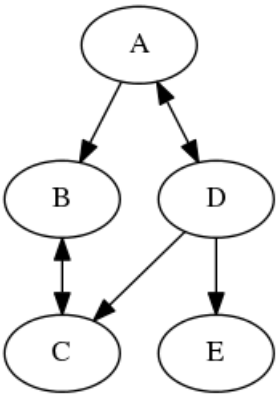
- Block Diagram:

1. Hash table:




Keys are the inputs taken from user which are used by the hash function to generate address for storage of the job/internship object.

2. Graph:




Graph gives the traversal direction i.e. the sequence in which the search filters are takes as input from the user.

The flow is Sector-->Domain-->Job/Internship-->Basic qualification-->Location

-->Working hours-->Other details.

Each node of the graph gives a specific value to hash function.

- Short note on Data Structure used:

Data Structure used is Graph and Hash-table.

Hash table ensures better synchronization also hash tables turn out to be more efficient than search trees. A simple array of linked list is used to build hash tables which makes it more efficient for searching a particular field.

Graph helps in better connectivity and shortest traversal.

File is used for storage and creating a database.

- Operations performed

1. Logic related to calculation of address of the hash table:

Each node in graph is given a specific value and then the address is calculated as = domain value + job internship value + work time value + work location value + education / experience value.

***