Implementation of Dict using Hashing

Shikha.N
1BM18CS149
25-11-20

```
Class HashTableEntry{
  public:
      int k;
      int v;
      HashTableEntry (int k, int v){
          this -> k = k;
          this -> v = v;
      }
};

Class HashMapTable{
   Private:
      Hash Table entry **t;
   public:
      HashMapTable(){
      t = new HashTableEntry * [Size];
      for (int i = 0; i< Size; i++)
            t[i]= NULL;
   }
   int hash Func (int k) {
        return k % Size;
   }
   void Insert (int k, int v){
        int h= HashFunc (k);
     while (t[h]!= NULL && t[h]->k!=k)
     {    h = hash func (h+1);
                 // Linear Probing
     }
     if (t[h]!=NULL)
          delete t[h];
     t[h] = new Hash Table Entry (K,V);
}
```

Shikha

```
int search(int k)
{ int h= hashFunc(k);
  while (t[h]!=NULL && t[h]->k!=k) {
            h = hashFunc(h+1);
  }
     if (t[h]==NULL)
              return -1;

     else
              return t[h]->v;
}
void deleteEle (int k) {
      int h = hashFunc(k);
     while (t[h] != NULL) {
          if (t[h] ->k == k)
                  break;
          h = hashfunc(h+1);
      }
      if (t[h] == NULL) {
          cout <<" No element found at key"<< k<<endl;
          return;

      } else { delete t[h];
                 cout << " element deleted \n"
              }
      }

  };
```

[illegible]