```
Language Fundamentals
+++++++++++++++++++++
  1. Identifiers
       A name in java program is called "identifier"
       The name can be a classname,interfacename,enumname,varaiblename,methodname
and label name.

class Test{
       public static void main(String[] args){
              int x = 10;
       }
}

Identifiers: Test,main,String,args,x

+++++++++++++++++++++
Rules for Identifiers
+++++++++++++++++++++
 Rule1: The allowed characters in java identifiers are
        a to z, A to Z,0 to 9,_,$

 Rule2: If we use any other character, then the program would result in
"CompileTime Error".

 Rule3: Identifiers should not start with digits
              eg: PWSKILLS123---> valid
                  123PWSKILLS---> invalid

 Rule4: java identifiers are case sensitive,as such Java language only is case
sensitive(JVM)
 class Test{
       int number = 10;
       int NUMBER = 100;
       int NuMbeR = 1000;
 }

 Rule5: There is no constraint of the length of the java identifiers,but it is not
recomended to take the length more than 15.
              eg: int physicsWallahAlakhPandeyJavaWithMicroservices = 100;

 Rule6: Reserve words or builtin words can't be used as "java identifiers".
       if we try to use then it would result in "CompiletimeError".
       int,if,float,else,while,for,do,..... => reserve words/built in words/keywords
       Inbuilt classname,interfacename,enumname can be used as a "identifier".
       Eventhough classname,interfacename,enumname can be used as an identifier, we
don't recomend.

eg::
class Test{
   public static void main(String[] args){
       int int=10; //CE
       int if =10; //CE
       int while=100;//CE
   }
}

class Test{
       public static void main(String[] args){
              int String =10;
```

```
        }
}
```

++
Identify which are valid and invalid identifiers
++
1. _$_ ===> valid
2. ca$h ==> valid
3.java2share=> valid
4. all@hands=> invalid
5.123abc => invalid
6.Total# => invalid
7.Int   => valid(not recomended)
8.Integer=> valid(not recomended)
9.int   => invalid(reserve word)
10.tot123=> valid