Lab assignment No 5 : Create User defined exception to check the following conditions and throw the exception if the criterion does not meet.

a. User stays in Pune/ Mumbai/ Bangalore / Chennai

b. User has 4-wheeler

City, Vehicle from the user and check for the conditions mentioned above. If any of the condition not met then throw the exception. If user does not enter proper input throw the exception

Aim: Use user defined exception handling to check invalid address and vehicle owned by person.

Description: In this task I firstly created person class accepting person's data. In that class I have declared one string array containing given addresses in lower case. During exception checking I have converted users address in to lower case string and thrown exception if no match found. Same thing is done to throw car exception.

Oop Concepts Used:

1)Exception Handling:
n exception is a problem that arises during the execution of a program. A C++ exception is a response to an exceptional circumstance that arises while a program is running, such as an attempt to divide by zero.

Exceptions provide a way to transfer control from one part of a program to another. C++ exception handling is built upon three keywords: **try, catch,** and **throw**.

- **throw** − A program throws an exception when a problem shows up. This is done using a **throw** keyword.

- **catch** − A program catches an exception with an exception handler at the place in a program where you want to handle the problem. The **catch** keyword indicates the catching of an exception.

- **try** − A **try** block identifies a block of code for which particular exceptions will be activated. It's followed by one or more catch blocks.

Assuming a block will raise an exception, a method catches an exception using a combination of the **try** and **catch** keywords. A try/catch block is placed around the code that might generate an exception.

Syntax:

```
try {
```

```
   // protected code
} catch( ExceptionName e1 ) {
   // catch block
} catch( ExceptionName e2 ) {
   // catch block
} catch( ExceptionName eN ) {
   // catch block
}
```

Throwing Exceptions

Exceptions can be thrown anywhere within a code block using **throw** statement. The operand of the throw statement determines a type for the exception and can be any expression and the type of the result of the expression determines the type of exception thrown.

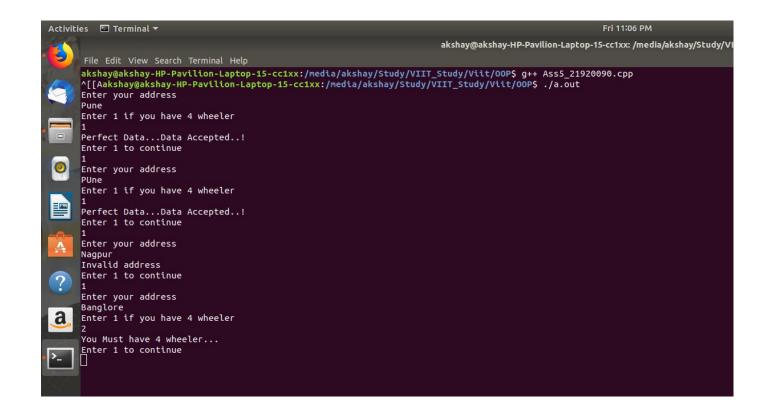Following is an example of throwing an exception when **dividing by zero condition** occurs −

```
double division(int a, int b) {
   if( b == 0 ) {
      throw "Division by zero condition!";
   }
   return (a/b);
}
```

Catching Exceptions

The **catch** block following the **try** block catches any exception. You can specify what type of exception you want to catch and this is determined by the exception declaration that appears in parentheses following the keyword catch.

```
try {
   // protected code
} catch( ExceptionName e ) {
  // code to handle ExceptionName exception
}
```

Output:

Conclusion:

Thus we have successfully implemented Exception handling.