# *Lab Test 01*

## Question 1: Error Detection [50]

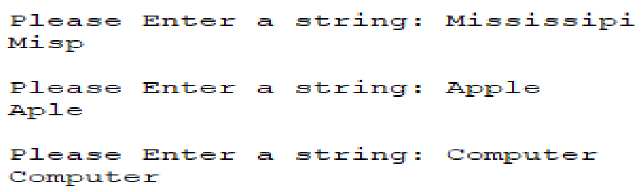
The given program is used to find the unique characters in a string. It expects a string as an input and prints the unique character in the input string as the expected output. The program has two user-defined methods/functions. The first is **getUniqueCharacter**, which expects a string as an argument. This method constructs the array of unique characters and returns it. The second method is **contains,** which checks weather a character exists on a given array or not.

The program contains some errors. Your task is to find the error, correct it and make the give code an executable program.

```java
public class Unique Character {

   public static void main(String[] args) {
      // TODO Auto-generated method stub
      System.out.print("Please Enter a string: ");
      Scanner keyboard = new Scanner(System.out);
      String input = keyboard.toString();
      char[] output = getUniqueCharacter(input);
      System.out.println(output);
   }
   public static char[] getUniqueCharacter(char s){
      int length = s.length();
      char[] str = s.toCharArray();
      char[] uniqueChar = new char[2];
      int counter = 0;
      for (int i = 0, i < length, i++) {
         if (!contains(str[i],uniqueChar)) {
               uniqueChar[counter] = str[i];
               counter++;
         }
      }
      return uniqueChar;
   }
   static boolean contains(char c, char[] array){
      for(char x:array){
         if (x == c) {
            return true;
             }
      }
      return false;
      }
}
```

### *Expected output:*



```
Please Enter a string: Mississipi
Misp

Please Enter a string: Apple
Aple

Please Enter a string: Computer
Computer
```

## Question 2: Programming [50]

Here is a program which reads the input from a file employee.txt and parses the file line-by-line which contains the information in the format **Employee_Name/Salary**. Finally, the program prints out the information of the employee (name and salary).

```
public class MainClass {

   public static void main(String[] args) throws Exception {
      // TODO Auto-generated method stub
      File address = new File ("/home/subik/Desktop/employee.txt");
      Scanner reader = new Scanner(address);
      Employee[] employeeList = new Employee[5];
      String[] info;
      int i=0;
      while(reader.hasNextLine()) {
         String line = reader.nextLine();
         info = line.split("/");
         employeeList[i] = new Employee(info[0], Integer.parseInt(info[1]));
         i++;
      }
      for(int j = 0; j<i; j++) {
         System.out.println(employeeList[j].toString());
      }
   }
}


public class Employee {

   private String name;
   private int salary;


   public Employee(String name, int salary) {
      this.name = name;
      this.salary = salary;
   }
   public String getName() {
      return name;
   }
   public void setName(String name) {
      this.name = name;
   }
   public int getSalary() {
      return salary;
   }
   public void setSalary(int salary) {
      this.salary = salary;
   }
   @Override
   public String toString() {
      return "Employee [name=" + name + ", salary=" + salary + "]";
   }
}
```

Your task is to modify the above program to do the following:

- Add another field to the Employee object to contain address.
- The program will then read through the file "employee.txt" and will parse the file line-by-line which will contains the information regarding the employee in the format of **Employee_Name/Salary/Address**.
- Your program should then calculate the average salary of the employees.
- Finally print out the details of the employee in the format:
    - (Employee_Name) / (Salary) / (Address) / (Absolute difference of salary with the average salary)

**BONUS:** [Doesn't exceed the max points]: Parse through the given employeeList file to determine the amount of the employee that will be in the array prior to making the employee array.

```
Slime/10000/Grand Forks
Chocobo/5000/Fargo
Murlocs/78000/Bismarck
Daedra/89000/Fargo
Eggman/145999/Grand Forks
```