CSC248 - FUNDAMENTALS OF DATA STRUCTURES
TREE: CLASS EXERCISE

NAME	**:**	MUHAMMAD AL-AMIN BIN MOHD ZAINI
STUDENT NO.	**:**	2018280578
GROUP	**:**	RCS1104B

QUESTION 1

A survey has been done related to the health status of Segamat community. Given the `HealthSurvey`, `TreeNode` and `BST` ADTs:

```
public class HealthSurvey
{
    private int serialNo;
    private int age;
    private char gender; //M - male & F - female
    private boolean medicine; //yes - true & no - false
    private boolean smoking; //yes - true & no - false
    private boolean exercise; //yes - true & no - false

    //definition of other methods
}

public class TreeNode
{
    TreeNode left;
    HealthSurvey data;
    TreeNode right;

    //definition of other methods
}

public class BST
{
    private TreeNode root;

    public BST() {...}
    public int cGenderSmoking(char gen) {...}
    public void displayHealthy() {...}

    //definition of other methods
}
```

(a) Write the definition of method `cGenderSmoking(char)` and its recursive method to count and return the number of respondents who smoke according to the gender given through parameter.

(b) Write the definition of method `displayHealthy()` and its recursive method to identify and display details of respondents who exercise and do not take any medicine.

(c) In the main program, get TEN (10) inputs to store respondents' details into a `BST` object and then call `cGenderSmoking(char)` and`displayHealthy()`.

```
a)
public char cGenderSmoking (char gender)
{
   If(gender==’M’)
      Return countSmokeM(root);
   Else if(gender==’F’)
      Return countSmokeF(root);
}

public int countSmokeM(TreeNode node)
{
   If(node==null)
      Return 0;
   Else if(node.data.getSmoking()==true && node.data.getGender()==M)
      Return 1 + countSmokeM(node.left) + countSmokeM(node.right);
   Else
      Return countSmokeM(node.left) + countSmokeM(node.right);

}

public int countSmokeF(TreeNode node)
{
   If(node==null)
      Return 0;
   Else if(node.data.getSmoking()==true && node.data.getGender()==F)
      Return 1 + countSmokeF(node.left) + countSmokeF(node.right);
   Else
      Return countSmokeF(node.left) + countSmokeF(node.right);

}
```

```
b)
public void displayHealthy()
{
   print(root);
}

Private void print(TreeNode node)
{
   If(node==null)
      Return;
   Else if(node.data.getExcercise()==true &&
node.date.getMedicine()==false)
      System.out.println(node.date+” “);
      print(node.left);
      print(node.right);
{
c)
BST healthTree=new BST()
Scanner scan=new Scanner(System.in);
For(int x=0 ; x<10 ; x++)
{
   System.out.println(“Enter Serial Number:”);
   Int serialNo=scan.nextInt();
   System.out.println(“Enter Age:”);
   Int age=scan.nextInt();
   System.out.println(“Enter Gender:”);
   Char gender=scan.next.CharAt(0);
   System.out.println(“Do You Take Medicine?”);
   Boolean medicine=scan.nextBoolean();
   System.out.println(“Do you smoke?”);
   Boolean smoking=scan.nextBoolean();
   System.out.println(“Do you exercise?”);
   Boolean exercise=scan.nextBoolean();

   HealthSurvey hs=new HealthSurvey(serialNo, age, gender, medicine,
smoking,exercise);
}
   System.out.println("\nEnter gender to count and return the number
of respondents who smoke by gender);
   Char gender_=scan.next.CharAt(0);

   System.out.println("\nNumber of respondents who smoke by
gender:"+healthTree.cGenderSmoking(gender_));

   System.out.println("\nDisplay details of respondents who exercise
and do not take any medicine.:");
        healthTree.displayHealthy();
```

QUESTION 2

Given the `Kindergarten`, `TreeNode` and `BSTKindergarten` ADTs:

```
public class Kindergarten
{
    private String name;
    private char gender;
    private int age;
    private String className; //Eg: 5 Red, 4 Green, 4 Red
    private String teacher;

    //definition of other methods
}

public class TreeNode
{
    TreeNode left;
    Kindergarten data;
    TreeNode right;

    //definition of other methods
}

public class BSTKindergarten
{
    private TreeNode root;

    public BSTKindergarten() {...}
    public void countGender(char gender) {...}
    public void displayChildren(String className) {...}

    //definition of other methods
}
```

(a) Write the definition of `countGender(char)` method and its recursive method to count and display the numbers of children that is specified through parameter according to the gender in the kindergarten.

(b) Write the definition of `displayChildren(String)` method to display the details of children from the class name that is specified through parameter according to the children name in ascending order.

(c) Assume that 20 objects of `Kindergarten` class have been inserted into a tree named `KinderTree`. In `main()` method, write Java statements to call the methods to display:

(i) The number of male students in the kindergarten

(ii) The children's details will be sorted according to the children name in ascending order for a specific class.

```
a)
public char countGender(char gender)
{
   If(gender==M)
      Return countM(root);
   Else if(gender==F)
      Return countF(root)
}

public int countM(TreeNode node)
{
   If(node==null)
      Return 0;
   Else if(node.data.getGender()==gender)
      Return 1 + countM(node.left) + countM(node.right);
   Else
      Return countM(node.left) + countM(node.right);
}

public int countF(TreeNode node)
{
   If(node==null)
      Return 0;
   Else if(node.data.getGender()==gender)
      Return 1 + countF(node.left) + countF(node.right);
   Else
      Return countF(node.left) + countF(node.right);
}


b)
public String displayChildren(className)
   {
      dChildren(root);
   }

   private void dChildren(TreeNode node)
   {
      if (node==null)
         return;
      else if(node.data.getClassName()==classname)
         dChildren(node.left);

      System.out.println(node.data);
      dChildren(node.right);
   }
```

```
c)
(i)
char gender_= M;
System.out.println("\nNumber of male students in the
kindergarten:"+KinderTree.countGender(gender_));

(ii)
System.out.println(“Enter class name :”);
String classname_=scan.nextLine();

System.out.println("\nDisplay children's details sorted according to
the children name in ascending order for a specific class:");
        KindergartenTree.displayChildren(className_);
```