```
GROUP 11 (ATTENDANCE REGISTER)
GROUP MEMBERS:
   MATAKALA HUDSON NGWENYA
   BISHONGA REVELATION
   SIAME MOSES



Class name: Main
Total number of tokens: 997
SPACE: 559
ASSIGN: 16
LBRACE: 9
LPAREN: 26
IMPORT: 1
MULTI_LINE_COMMENT: 1
VOID: 1
SINGLE_LINE_COMMENT: 6
EOF: 1
STATIC: 1
WINDOWS_EOL: 60
RBRACKET: 15
FOR: 4
IF: 2
LBRACKET: 15
LT: 4
RBRACE: 9
PLUS: 8
SEMICOLON: 35
INT: 7
DECR: 1
CLASS: 1
INTEGER_LITERAL: 6
RPAREN: 26
PUBLIC: 2
IDENTIFIER: 119
ELSE: 2
STRING_LITERAL: 13
INCR: 6
NEW: 5
DOT: 36

ParserMainAST
Class name: Main
Class AST:
ClassOrInterfaceDeclaration: public class Main {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        //Variable to store the lenght of the register array
        System.out.println("Enter the number of members in the
attendance register: ");
        int x = input.nextInt();
        // Initiaise array and display prompt to enter the names
        String[] names = new String[x];
        System.out.println("Enter the names: ");
        // It gets the names from the console and input into the array
        for (int i = 0; i < names.length; i++) {
            Scanner input2 = new Scanner(System.in);
            names[i] = input2.nextLine();
        }
        /*  prints the names of the array
        System.out.println("The following are the names in the
register: ");
        for(int i = 0; i < x; i++){
           System.out.println(names[i]);

        } */
        String[] presentArray = new String[names.length];
        String[] absentArray = new String[names.length];
        int presentIndex = 0;
        int absentIndex = 0;
        // Prompt the user to classify each element as absent or
present
        for (int i = 0; i < names.length; i++) {
            System.out.print("Is " + names[i] + " present? (Y/N): ");
            String response = input.nextLine().toUpperCase();
            System.out.println();
            if (response.equals("Y")) {
                presentArray[presentIndex] = names[i];
                presentIndex++;
            } else if (response.equals("N")) {
                absentArray[absentIndex] = names[i];
                absentIndex++;
            } else {
                System.out.println("Invalid response. Please enter Y or
N.");
                i--;
            }
        }
        // Print the present array
        System.out.println("Present Members: " + " " + presentIndex);
        for (int i = 0; i < presentIndex; i++) {
            System.out.println(presentArray[i] + " ");
        }
        System.out.println();
        // Print the absent array
        System.out.println("Absent Members: " + " " + absentIndex);
        for (int i = 0; i < absentIndex; i++) {
            System.out.println(absentArray[i] + " ");
        }
        System.out.println();
    }
}
  Modifier: public
  SimpleName: Main
  MethodDeclaration: public static void main(String[] args) {
    Scanner input = new Scanner(System.in);
    //Variable to store the lenght of the register array
    System.out.println("Enter the number of members in the attendance
register: ");
    int x = input.nextInt();
    // Initiaise array and display prompt to enter the names
    String[] names = new String[x];
    System.out.println("Enter the names: ");
    // It gets the names from the console and input into the array
    for (int i = 0; i < names.length; i++) {
        Scanner input2 = new Scanner(System.in);
        names[i] = input2.nextLine();
    }
    /*  prints the names of the array
        System.out.println("The following are the names in the
register: ");
        for(int i = 0; i < x; i++){
           System.out.println(names[i]);

        } */
    String[] presentArray = new String[names.length];
    String[] absentArray = new String[names.length];
    int presentIndex = 0;
    int absentIndex = 0;
    // Prompt the user to classify each element as absent or present
    for (int i = 0; i < names.length; i++) {
        System.out.print("Is " + names[i] + " present? (Y/N): ");
        String response = input.nextLine().toUpperCase();
        System.out.println();
        if (response.equals("Y")) {
            presentArray[presentIndex] = names[i];
            presentIndex++;
        } else if (response.equals("N")) {
            absentArray[absentIndex] = names[i];
            absentIndex++;
        } else {
            System.out.println("Invalid response. Please enter Y or
N.");
            i--;
        }
    }
    // Print the present array
    System.out.println("Present Members: " + " " + presentIndex);
    for (int i = 0; i < presentIndex; i++) {
        System.out.println(presentArray[i] + " ");
    }
    System.out.println();
    // Print the absent array
    System.out.println("Absent Members: " + " " + absentIndex);
    for (int i = 0; i < absentIndex; i++) {
        System.out.println(absentArray[i] + " ");
    }
    System.out.println();
}
    Modifier: public
    Modifier: static
    SimpleName: main
    Parameter: String[] args
      ArrayType: String[]
        ClassOrInterfaceType: String
          SimpleName: String
      SimpleName: args
    VoidType: void
    BlockStmt: {
    Scanner input = new Scanner(System.in);
    //Variable to store the lenght of the register array
    System.out.println("Enter the number of members in the attendance
register: ");
    int x = input.nextInt();
    // Initiaise array and display prompt to enter the names
    String[] names = new String[x];
    System.out.println("Enter the names: ");
    // It gets the names from the console and input into the array
    for (int i = 0; i < names.length; i++) {
        Scanner input2 = new Scanner(System.in);
        names[i] = input2.nextLine();
    }
    /*  prints the names of the array
        System.out.println("The following are the names in the
register: ");
        for(int i = 0; i < x; i++){
           System.out.println(names[i]);

        } */
    String[] presentArray = new String[names.length];
    String[] absentArray = new String[names.length];
    int presentIndex = 0;
    int absentIndex = 0;
    // Prompt the user to classify each element as absent or present
    for (int i = 0; i < names.length; i++) {
        System.out.print("Is " + names[i] + " present? (Y/N): ");
        String response = input.nextLine().toUpperCase();
        System.out.println();
        if (response.equals("Y")) {
            presentArray[presentIndex] = names[i];
            presentIndex++;
        } else if (response.equals("N")) {
            absentArray[absentIndex] = names[i];
            absentIndex++;
        } else {
            System.out.println("Invalid response. Please enter Y or
N.");
            i--;
        }
    }
    // Print the present array
    System.out.println("Present Members: " + " " + presentIndex);
    for (int i = 0; i < presentIndex; i++) {
        System.out.println(presentArray[i] + " ");
    }
    System.out.println();
    // Print the absent array
    System.out.println("Absent Members: " + " " + absentIndex);
    for (int i = 0; i < absentIndex; i++) {
        System.out.println(absentArray[i] + " ");
    }
    System.out.println();
}
      ExpressionStmt: Scanner input = new Scanner(System.in);
        VariableDeclarationExpr: Scanner input = new Scanner(System.in)
          VariableDeclarator: input = new Scanner(System.in)
            ClassOrInterfaceType: Scanner
              SimpleName: Scanner
            SimpleName: input
            ObjectCreationExpr: new Scanner(System.in)
              ClassOrInterfaceType: Scanner
                SimpleName: Scanner
              FieldAccessExpr: System.in
                NameExpr: System
                  SimpleName: System
                SimpleName: in
      ExpressionStmt: //Variable to store the lenght of the register
array
System.out.println("Enter the number of members in the attendance
register: ");
        MethodCallExpr: System.out.println("Enter the number of members
in the attendance register: ")
          FieldAccessExpr: System.out
            NameExpr: System
              SimpleName: System
            SimpleName: out
          SimpleName: println
          StringLiteralExpr: "Enter the number of members in the
attendance register: "
      ExpressionStmt: int x = input.nextInt();
        VariableDeclarationExpr: int x = input.nextInt()
          VariableDeclarator: x = input.nextInt()
            PrimitiveType: int
            SimpleName: x
            MethodCallExpr: input.nextInt()
              NameExpr: input
                SimpleName: input
              SimpleName: nextInt
      ExpressionStmt: // Initiaise array and display prompt to enter
the names
String[] names = new String[x];
        VariableDeclarationExpr: String[] names = new String[x]
          VariableDeclarator: names = new String[x]
            ArrayType: String[]
              ClassOrInterfaceType: String
                SimpleName: String
            SimpleName: names
            ArrayCreationExpr: new String[x]
              ClassOrInterfaceType: String
                SimpleName: String
              ArrayCreationLevel: [x]
                NameExpr: x
                  SimpleName: x
      ExpressionStmt: System.out.println("Enter the names: ");
        MethodCallExpr: System.out.println("Enter the names: ")
          FieldAccessExpr: System.out
            NameExpr: System
              SimpleName: System
            SimpleName: out
          SimpleName: println
          StringLiteralExpr: "Enter the names: "
      ForStmt: // It gets the names from the console and input into the
array
for (int i = 0; i < names.length; i++) {
    Scanner input2 = new Scanner(System.in);
    names[i] = input2.nextLine();
}
```

```
        VariableDeclarationExpr: int i = 0
          VariableDeclarator: i = 0
            PrimitiveType: int
            SimpleName: i
            IntegerLiteralExpr: 0
        BinaryExpr: i < names.length
          NameExpr: i
            SimpleName: i
          FieldAccessExpr: names.length
            NameExpr: names
              SimpleName: names
            SimpleName: length
        UnaryExpr: i++
          NameExpr: i
            SimpleName: i
        BlockStmt: {
    Scanner input2 = new Scanner(System.in);
    names[i] = input2.nextLine();
}
          ExpressionStmt: Scanner input2 = new Scanner(System.in);
            VariableDeclarationExpr: Scanner input2 = new
Scanner(System.in)
              VariableDeclarator: input2 = new Scanner(System.in)
                ClassOrInterfaceType: Scanner
                  SimpleName: Scanner
                SimpleName: input2
                ObjectCreationExpr: new Scanner(System.in)
                  ClassOrInterfaceType: Scanner
                    SimpleName: Scanner
                  FieldAccessExpr: System.in
                    NameExpr: System
                      SimpleName: System
                    SimpleName: in
          ExpressionStmt: names[i] = input2.nextLine();
            AssignExpr: names[i] = input2.nextLine()
              ArrayAccessExpr: names[i]
                NameExpr: names
                  SimpleName: names
                NameExpr: i
                  SimpleName: i
              MethodCallExpr: input2.nextLine()
                NameExpr: input2
                  SimpleName: input2
                SimpleName: nextLine
      ExpressionStmt: /*  prints the names of the array
        System.out.println("The following are the names in the
register: ");
        for(int i = 0; i < x; i++){
           System.out.println(names[i]);

        } */
String[] presentArray = new String[names.length];
        VariableDeclarationExpr: String[] presentArray = new
String[names.length]
          VariableDeclarator: presentArray = new String[names.length]
            ArrayType: String[]
              ClassOrInterfaceType: String
                SimpleName: String
            SimpleName: presentArray
            ArrayCreationExpr: new String[names.length]
              ClassOrInterfaceType: String
                SimpleName: String
              ArrayCreationLevel: [names.length]
                FieldAccessExpr: names.length
                  NameExpr: names
                    SimpleName: names
                  SimpleName: length
      ExpressionStmt: String[] absentArray = new String[names.length];
        VariableDeclarationExpr: String[] absentArray = new
String[names.length]
          VariableDeclarator: absentArray = new String[names.length]
            ArrayType: String[]
              ClassOrInterfaceType: String
                SimpleName: String
            SimpleName: absentArray
            ArrayCreationExpr: new String[names.length]
              ClassOrInterfaceType: String
                SimpleName: String
              ArrayCreationLevel: [names.length]
                FieldAccessExpr: names.length
                  NameExpr: names
                    SimpleName: names
                  SimpleName: length
      ExpressionStmt: int presentIndex = 0;
        VariableDeclarationExpr: int presentIndex = 0
          VariableDeclarator: presentIndex = 0
            PrimitiveType: int
            SimpleName: presentIndex
            IntegerLiteralExpr: 0
      ExpressionStmt: int absentIndex = 0;
        VariableDeclarationExpr: int absentIndex = 0
          VariableDeclarator: absentIndex = 0
            PrimitiveType: int
            SimpleName: absentIndex
            IntegerLiteralExpr: 0
      ForStmt: // Prompt the user to classify each element as absent or
present
for (int i = 0; i < names.length; i++) {
    System.out.print("Is " + names[i] + " present? (Y/N): ");
    String response = input.nextLine().toUpperCase();
    System.out.println();
    if (response.equals("Y")) {
        presentArray[presentIndex] = names[i];
        presentIndex++;
    } else if (response.equals("N")) {
        absentArray[absentIndex] = names[i];
        absentIndex++;
    } else {
        System.out.println("Invalid response. Please enter Y or N.");
        i--;
    }
}
        VariableDeclarationExpr: int i = 0
          VariableDeclarator: i = 0
            PrimitiveType: int
            SimpleName: i
            IntegerLiteralExpr: 0
        BinaryExpr: i < names.length
          NameExpr: i
            SimpleName: i
          FieldAccessExpr: names.length
            NameExpr: names
              SimpleName: names
            SimpleName: length
        UnaryExpr: i++
          NameExpr: i
            SimpleName: i
        BlockStmt: {
    System.out.print("Is " + names[i] + " present? (Y/N): ");
    String response = input.nextLine().toUpperCase();
    System.out.println();
    if (response.equals("Y")) {
        presentArray[presentIndex] = names[i];
        presentIndex++;
    } else if (response.equals("N")) {
        absentArray[absentIndex] = names[i];
        absentIndex++;
    } else {
        System.out.println("Invalid response. Please enter Y or N.");
        i--;
    }
}
          ExpressionStmt: System.out.print("Is " + names[i] + "
present? (Y/N): ");
            MethodCallExpr: System.out.print("Is " + names[i] + "
present? (Y/N): ")
              FieldAccessExpr: System.out
                NameExpr: System
                  SimpleName: System
                SimpleName: out
              SimpleName: print
              BinaryExpr: "Is " + names[i] + " present? (Y/N): "
                BinaryExpr: "Is " + names[i]
                  StringLiteralExpr: "Is "
                  ArrayAccessExpr: names[i]
                    NameExpr: names
                      SimpleName: names
                    NameExpr: i
                      SimpleName: i
                StringLiteralExpr: " present? (Y/N): "
          ExpressionStmt: String response =
input.nextLine().toUpperCase();
            VariableDeclarationExpr: String response =
input.nextLine().toUpperCase()
              VariableDeclarator: response =
input.nextLine().toUpperCase()
                ClassOrInterfaceType: String
                  SimpleName: String
                SimpleName: response
                MethodCallExpr: input.nextLine().toUpperCase()
                  MethodCallExpr: input.nextLine()
                    NameExpr: input
                      SimpleName: input
                    SimpleName: nextLine
              SimpleName: toUpperCase
          ExpressionStmt: System.out.println();
            MethodCallExpr: System.out.println()
              FieldAccessExpr: System.out
                NameExpr: System
                  SimpleName: System
                SimpleName: out
              SimpleName: println
          IfStmt: if (response.equals("Y")) {
    presentArray[presentIndex] = names[i];
    presentIndex++;
} else if (response.equals("N")) {
    absentArray[absentIndex] = names[i];
    absentIndex++;
} else {
    System.out.println("Invalid response. Please enter Y or N.");
    i--;
}
            MethodCallExpr: response.equals("Y")
              NameExpr: response
                SimpleName: response
              SimpleName: equals
              StringLiteralExpr: "Y"
            BlockStmt: {
    presentArray[presentIndex] = names[i];
    presentIndex++;
}
              ExpressionStmt: presentArray[presentIndex] = names[i];
                AssignExpr: presentArray[presentIndex] = names[i]
                  ArrayAccessExpr: presentArray[presentIndex]
                    NameExpr: presentArray
                      SimpleName: presentArray
                    NameExpr: presentIndex
                      SimpleName: presentIndex
                  ArrayAccessExpr: names[i]
                    NameExpr: names
                      SimpleName: names
                    NameExpr: i
                      SimpleName: i
              ExpressionStmt: presentIndex++;
                UnaryExpr: presentIndex++
                  NameExpr: presentIndex
                    SimpleName: presentIndex
            IfStmt: if (response.equals("N")) {
    absentArray[absentIndex] = names[i];
    absentIndex++;
} else {
    System.out.println("Invalid response. Please enter Y or N.");
    i--;
}
              MethodCallExpr: response.equals("N")
                NameExpr: response
                  SimpleName: response
                SimpleName: equals
                StringLiteralExpr: "N"
              BlockStmt: {
    absentArray[absentIndex] = names[i];
    absentIndex++;
}
                ExpressionStmt: absentArray[absentIndex] = names[i];
                  AssignExpr: absentArray[absentIndex] = names[i]
                    ArrayAccessExpr: absentArray[absentIndex]
                      NameExpr: absentArray
                        SimpleName: absentArray
                      NameExpr: absentIndex
                        SimpleName: absentIndex
                    ArrayAccessExpr: names[i]
                      NameExpr: names
                        SimpleName: names
                      NameExpr: i
                        SimpleName: i
                ExpressionStmt: absentIndex++;
                  UnaryExpr: absentIndex++
                    NameExpr: absentIndex
                      SimpleName: absentIndex
              BlockStmt: {
    System.out.println("Invalid response. Please enter Y or N.");
    i--;
}
                ExpressionStmt: System.out.println("Invalid response.
Please enter Y or N.");
                  MethodCallExpr: System.out.println("Invalid response.
Please enter Y or N.")
                    FieldAccessExpr: System.out
                      NameExpr: System
                        SimpleName: System
                      SimpleName: out
                    SimpleName: println
                    StringLiteralExpr: "Invalid response. Please enter
Y or N."
                ExpressionStmt: i--;
                  UnaryExpr: i--
                    NameExpr: i
                      SimpleName: i
      ExpressionStmt: // Print the present array
System.out.println("Present Members: " + " " + presentIndex);
        MethodCallExpr: System.out.println("Present Members: " + " " +
presentIndex)
          FieldAccessExpr: System.out
            NameExpr: System
              SimpleName: System
            SimpleName: out
          SimpleName: println
          BinaryExpr: "Present Members: " + " " + presentIndex
            BinaryExpr: "Present Members: " + " "
              StringLiteralExpr: "Present Members: "
              StringLiteralExpr: " "
            NameExpr: presentIndex
              SimpleName: presentIndex
      ForStmt: for (int i = 0; i < presentIndex; i++) {
    System.out.println(presentArray[i] + " ");
}
        VariableDeclarationExpr: int i = 0
          VariableDeclarator: i = 0
            PrimitiveType: int
            SimpleName: i
            IntegerLiteralExpr: 0
        BinaryExpr: i < presentIndex
          NameExpr: i
            SimpleName: i
          NameExpr: presentIndex
            SimpleName: presentIndex
        UnaryExpr: i++
          NameExpr: i
            SimpleName: i
        BlockStmt: {
    System.out.println(presentArray[i] + " ");
}
          ExpressionStmt: System.out.println(presentArray[i] + " ");
            MethodCallExpr: System.out.println(presentArray[i] + " ")
              FieldAccessExpr: System.out
                NameExpr: System
                  SimpleName: System
                SimpleName: out
              SimpleName: println
              BinaryExpr: presentArray[i] + " "
                ArrayAccessExpr: presentArray[i]
                  NameExpr: presentArray
                    SimpleName: presentArray
                  NameExpr: i
                    SimpleName: i
                StringLiteralExpr: " "
      ExpressionStmt: System.out.println();
        MethodCallExpr: System.out.println()
          FieldAccessExpr: System.out
            NameExpr: System
              SimpleName: System
            SimpleName: out
          SimpleName: println
      ExpressionStmt: // Print the absent array
System.out.println("Absent Members: " + " " + absentIndex);
        MethodCallExpr: System.out.println("Absent Members: " + " " +
absentIndex)
          FieldAccessExpr: System.out
            NameExpr: System
              SimpleName: System
            SimpleName: out
          SimpleName: println
          BinaryExpr: "Absent Members: " + " " + absentIndex
            BinaryExpr: "Absent Members: " + " "
              StringLiteralExpr: "Absent Members: "
              StringLiteralExpr: " "
            NameExpr: absentIndex
              SimpleName: absentIndex
      ForStmt: for (int i = 0; i < absentIndex; i++) {
    System.out.println(absentArray[i] + " ");
}
        VariableDeclarationExpr: int i = 0
          VariableDeclarator: i = 0
            PrimitiveType: int
            SimpleName: i
            IntegerLiteralExpr: 0
        BinaryExpr: i < absentIndex
          NameExpr: i
```

```
            SimpleName: i
          NameExpr: absentIndex
            SimpleName: absentIndex
        UnaryExpr: i++
          NameExpr: i
            SimpleName: i
        BlockStmt: {
    System.out.println(absentArray[i] + " ");
}
          ExpressionStmt: System.out.println(absentArray[i] + " ");
            MethodCallExpr: System.out.println(absentArray[i] + " ")
              FieldAccessExpr: System.out
                NameExpr: System
                  SimpleName: System
                SimpleName: out
              SimpleName: println
              BinaryExpr: absentArray[i] + " "
                ArrayAccessExpr: absentArray[i]
                  NameExpr: absentArray
                    SimpleName: absentArray
                  NameExpr: i
                    SimpleName: i
                StringLiteralExpr: " "
      ExpressionStmt: System.out.println();
        MethodCallExpr: System.out.println()
          FieldAccessExpr: System.out
            NameExpr: System
              SimpleName: System
            SimpleName: out
          SimpleName: println
      BlockComment: /*  prints the names of the array
        System.out.println("The following are the names in the
register: ");
        for(int i = 0; i < x; i++){
           System.out.println(names[i]);

        } */
```