```
##########################################

Student Name = Jitong Ding

##########################################



CSE017 Grading sheet for Jitong Ding

Homework Assignment WordMath

Total points                          maximum: 100

Completeness: All class/methods included (40)   [ 40 ]

Compilation: Program compiles (20)              [ 20 ]

Execution: Program executes properly (30)       [ 26 ]

Style: Program obeys style rules (10)           [ 10 ]

Subtotal                                          96

Late Penalty                                    [    ]

Total Points                                      96


##########################################
##########################################
```

-4 total for execution

```
/*
CSE 17
Jitong Ding
jid221
Program #3 DEADLINE: November 7, 2018
Program Description: Conversational Arithmetic
*/

public class SubtractionExpression extends Expression{
  /** Construct a new SubtractionExpression with leftOp and rightOp and throws I
nvalidQuestionException */
  public SubtractionExpression(String leftOp, String rightOp) throws InvalidQues
tionException{
    super(leftOp,rightOp);
}
  /** A method to return the calculating result*/
  public int evaluate(){
    return (leftOpInt - rightOpInt);
  }
}
```

```
/*
CSE 17
Jitong Ding
jid221
Program #3 DEADLINE: November 7, 2018
Program Description: Conversational Arithmetic
*/

import java.util.ArrayList;
import java.util.Scanner;

public class WordMath{
  /** Public static final data field */
  public static final int ADDITION =0;
  public static final int SUBTRACTION =1;
  public static final int MULTIPLICATION =2;
  public static final int DIVISION =3;

   /** A method to attempt to match question to one of the entries in templates.
 For the first match
found, parses the question using the template and returns the resulting Expressi
on. If no matching template is
found, throws an InvalidQuestionException.*/
  public static Expression parseQuestion(String question, ArrayList<QuestionTemp
late> templates) throws InvalidQuestionException{
    for(int i = 0; i<templates.size();++i){
      try{
        Expression temp = templates.get(i).parseQuestion(question);
        return temp;
      }
      catch(InvalidQuestionException ex){
        continue;
      }
    }
      throw new InvalidQuestionException();

  }

   /** The main method */
  public static void main(String[] args){
    ArrayList<QuestionTemplate> templateList = new ArrayList<>();
    templateList.add(new QuestionTemplate("What is","plus","?",ADDITION));
    templateList.add(new QuestionTemplate("Tell me the sum of","and","?",ADDITION));
    templateList.add(new QuestionTemplate("What do I get if I add","and","together?",ADDITI
ON));
    templateList.add(new QuestionTemplate("What is","minus","?",SUBTRACTION));
    templateList.add(new QuestionTemplate("","less","is what?",SUBTRACTION));
    templateList.add(new QuestionTemplate("What is","times","?",MULTIPLICATION));
    templateList.add(new QuestionTemplate("What do I get when I multiply","and","?",MULTI
PLICATION));
    templateList.add(new QuestionTemplate("What is","divided by","?",DIVISION));
    Scanner scan = new Scanner(System.in);
    System.out.println("Enter a question:");
    String questionWords = scan.nextLine();
    try{
      Expression temp = parseQuestion(questionWords,templateList);
      System.out.println("The answer is "+temp.evaluateInWords());
  }
    catch(InvalidQuestionException ex){
      System.out.println("I'm sorry, I don't understand the question. Please rerun and rephrase your questi
on.");
}
  }

}
```

```
/*
CSE 17
Jitong Ding
jid221
Program #3 DEADLINE: November 7, 2018
Program Description: Conversational Arithmetic
*/

public class AdditionExpression extends Expression{

  /** Construct a new AdditionExpression with leftOp and rightOp and throws InvalidQuestionException */
  public AdditionExpression(String leftOp, String rightOp) throws InvalidQuestionException{
    super(leftOp,rightOp);
  }
  /** A method to return the calculating result*/
  public int evaluate(){
    return (leftOpInt + rightOpInt);
  }
}

```

```
/*
CSE 17
Jitong Ding
jid221
Program #3 DEADLINE: November 7, 2018
Program Description: Conversational Arithmetic
*/

public class DivisionExpression extends Expression{
  /** Construct a new DivisionExpression with leftOp and rightOp and throws InvalidQuestionException */
  public DivisionExpression(String leftOp, String rightOp) throws InvalidQuestionException{
     super(leftOp,rightOp);
  }
  /** A method to return the calculating result*/
  public int evaluate(){
    return (leftOpInt / rightOpInt);
  }
}
```

```
/*
CSE 17
Jitong Ding
jid221
Program #3 DEADLINE: November 7, 2018
Program Description: Conversational Arithmetic
*/

/** A abstract class */
public abstract class Expression{

  /** Protected data field */
  protected int leftOpInt;
  protected int rightOpInt;
  /** A single NumberConvertor that can be used by any object that descends from Expression. */
  public static NumberConvertor numConvertor = new NumberConvertor();

  /** Construct a new Expression with leftOp and rightOp and throws InvalidQuestionException */
  public Expression(String leftOp, String rightOp) throws InvalidQuestionException{
      if(numConvertor.toNumber(leftOp)==-1 || numConvertor.toNumber(rightOp) == -1){
        throw new InvalidQuestionException();
}
      else{
        leftOpInt = numConvertor.toNumber(leftOp);
        rightOpInt = numConvertor.toNumber(rightOp);
}
  }

  /** An abstract method to perform the arithmetic operation using the left
and right operands and return the result */
  public abstract int evaluate();

  /** An method to perform the arithmetic operation and return the
numeric result expressed as words. */
  public String evaluateInWords(){
     NumberConvertor a = new NumberConvertor();
     return a.toWords(evaluate());
  }


}
```

```
/*
CSE 17
Jitong Ding
jid221
Program #3 DEADLINE: November 7, 2018
Program Description: Conversational Arithmetic
*/

public class InvalidQuestionException extends Exception{

  /** Construct a new InvalidQuestionException */
  public InvalidQuestionException(){
    super("Question not understood");
}
}
```

```
/*
CSE 17
Jitong Ding
jid221
Program #3 DEADLINE: November 7, 2018
Program Description: Conversational Arithmetic
*/

public class MultiplicationExpression extends Expression{
  /** Construct a new MultiplicationExpression with leftOp and rightOp and throw
s InvalidQuestionException */
  public MultiplicationExpression(String leftOp, String rightOp) throws InvalidQ
uestionException{
    super(leftOp,rightOp);
}
  /** A method to return the calculating result*/
  public int evaluate(){
    return (leftOpInt * rightOpInt);
  }
}
```

```
/*
CSE 17
Jitong Ding
jid221
Program #3 DEADLINE: November 7, 2018
Program Description: Conversational Arithmetic
*/

import java.util.ArrayList;

public class NumberConvertor{

  /** A private constant containing the word forms for the
numbers 0 through 19, inclusive.*/
  private static final String[] FIRST_NUMBERS = new String[]
  {"zero","one","two","three","four","five","six","seven","eight","nine","ten","eleven",
    "twelve","thirteen","fourteen","fifteen","sixteen","seventeen","eighteen","nineteen"};
  /** A private constant containing the word forms for all of
the multiples of 10 between 20 and 90 inclusive.*/
  private static final String[] TENS_NUMBERS = new String[]{"twenty","thirty","forty"
,"fifty",
  "sixty","seventy","eighty","ninety"};
  /** A private ArrayList containing NumberWords for 0-
19 and the multiples of 10 from 20-90.*/
  private ArrayList<NumberWord> numbersList = new ArrayList<>();

  /** Construct a new NumberConvertor initializing numbersList */
  public NumberConvertor(){
    for(int i =0; i<FIRST_NUMBERS.length;++i){
      numbersList.add(new NumberWord(i,FIRST_NUMBERS[i]));
    }
    for(int j =0; j<TENS_NUMBERS.length;++j){
      numbersList.add(new NumberWord((j+2)*10,TENS_NUMBERS[j]));
    }
    java.util.Collections.sort(numbersList);
}

  /** A method to use binary search to look it up and return the associated int
valueGiven a single number word,
    * by giving a single number word. If not found, returns -1.*/
   private int lookupValueOfWord(String numberWord){
     int low = 0;
     int high = numbersList.size()-1;

     while(high>=low){
       int mid = (high+low)/2;
       if(numberWord.compareToIgnoreCase(numbersList.get(mid).getInWords())<0){
         high = mid -1;
       }
       else if(numberWord.equalsIgnoreCase(numbersList.get(mid).getInWords())){
         return numbersList.get(mid).getValue();
       }
       else{
         low = mid +1;
  }
     }
     return -1;
}

   /** A method to use the numbersList, converts the words for any number betwee
n 0 and 99 inclusive into
its integer equivalent. Returns -1 for any number that it does not recognize.*/
   public int toNumber(String numberWords){
     NumberConvertor numConvertor = new NumberConvertor();
     if(numberWords.contains("-")){
       String[] numWords = numberWords.split("-",2);
       int a = numConvertor.lookupValueOfWord(numWords[0]);
       int b = numConvertor.lookupValueOfWord(numWords[1]);
       return (a+b);
     }
     else{
       return numConvertor.lookupValueOfWord(numberWords);
     }
```



```
    }

   /** A method to converts a numeric value between 0 to 99 intothe equivalent w
ord(s). If the number is
outside the range, returns not a number I can put into words (value)*/
   public String toWords(int value){
     NumberConvertor aNumConvertor = new NumberConvertor();
     if(value>0 && value<100){
       if(value>=0 && value<=19){
         return FIRST_NUMBERS[value];
     }
       else if (value%10==0 && value!=10){
         return TENS_NUMBERS[(value/10)-2];
     }
       else{
         String words = "-";
         for(int i = 0; i<2;++i){
           if(i==1){
             return words = TENS_NUMBERS[value-2] + words;
           }
           int remain = value%10;
           words = words + FIRST_NUMBERS[remain];
           value/=10;
         }
       }
}
     return "not a number I can put into words ("+ value+")";
   }

}
```

```
/*
CSE 17
Jitong Ding
jid221
Program #3 DEADLINE: November 7, 2018
Program Description: Conversational Arithmetic
*/


public class NumberWord implements Comparable<NumberWord>{

  /** Private data field */
  private int value;
  private String inWords;

  /** Construct a new NumberWord with value and inWords */
  public NumberWord(int value,String inWords){
    this.value= value;
    this.inWords = inWords;
  }

  /** A method to return value*/
  public int getValue(){
    return value;
  }

  /** A method to return inWords*/
  public String getInWords(){
    return inWords;
  }

  /** A method to compare two String*/
  public int compareTo(NumberWord obj){
    return inWords.compareToIgnoreCase(obj.getInWords());
  }

}
```

```
/*
CSE 17
Jitong Ding
jid221
Program #3 DEADLINE: November 7, 2018
Program Description: Conversational Arithmetic
*/


public class QuestionTemplate{

  /** Private data field */
  private String preText;
  private String middleText;
  private String endText;
  private int operation;

   /** Construct a new QuestionTemplate with preText, middleText, endText, operation */
  public QuestionTemplate(String preText, String middleText, String endText, int operation){
    this.preText = preText;
    this.middleText= middleText;
    this.endText = endText;
    this.operation = operation;
  }

  /** A method to returns true if question matches the template.*/
  public boolean isMatch(String question){
    String theQuestion = question.trim();
    int pre = theQuestion.indexOf(preText);
    int mid = theQuestion.indexOf(middleText);
    int end = theQuestion.indexOf(endText);
    if (pre==-1 || mid ==-1 || end==-1){
      return false;
    }
    if (end>mid&&mid>pre)
      return true;
    return false;
  }

  /** A method is given a question for which isMatch() is known to be true, and returns the equivalent subtype of
    * Expression for the question. The left operand for the expression is assumed to be the string between the pretext and the
middleText. The right operand is the string between middleText and endText. If either operand is invalid, then throw an
InvalidQuestionException.*/
  public Expression parseQuestion(String question) throws InvalidQuestionException{
    String theQuestion = question.trim();
    int pre = theQuestion.indexOf(preText);
    int mid = theQuestion.indexOf(middleText);
    int end = theQuestion.indexOf(endText);
    String preText1;
    String middleText1;
    String endText1;
    QuestionTemplate template;
    if(pre==0 && mid>0 && end>0){
      preText1 = theQuestion.substring(0,preText.length());
      middleText1 = theQuestion.substring(mid,(mid+middleText.length()));;
      endText1 = theQuestion.substring(end,(end+endText.length()));
      template = new QuestionTemplate(preText1,middleText1,endText1,operation);
      if(template.isMatch(question)){
        String Num1;
        String Num2;
        if(preText.length()==0){
          Num1 = theQuestion.substring(0,mid-1);
          Num2 = theQuestion.substring(mid+middleText.length()+1,end-1);
        }
        else{
          String theNum2;
          Num1 = theQuestion.substring(preText.length()+1,mid-1);
```



```
          theNum2 = theQuestion.substring(mid+middleText.length()+1,end);
          if(theNum2.charAt(theNum2.length()-1)==' '){
            Num2 = theNum2.substring(0,theNum2.length()-1);
          }
          else{
            Num2 = theNum2;
        }
        }
        if(operation==0){
          Expression addition = new AdditionExpression(Num1,Num2);
          return addition;
      }
        else if(operation==1){
          Expression subtraction = new SubtractionExpression(Num1,Num2);
          return subtraction;
      }
        else if(operation==2){
          Expression multiplication = new MultiplicationExpression(Num1,Num2);
          return multiplication;
      }
        else if(operation==3){
          Expression division = new DivisionExpression(Num1,Num2);
          return division;
      }
    }
    }
      throw new InvalidQuestionException();
}
  }
```

```
##########################################
##########################################

############ Compiled Result ###############

Source Code Compilation:

Submitted Code Compilation:

Test Cases Compilation:

##########################################

##########################################
############ Execution Result ##############
##########################################

##########################################
Regular Test

1

Question:

What is twenty-one plus eleven?
Expectated:
the answer is thirty-two
Output:

Enter a question:

The answer is thirty-two

2

Question:

Tell me the sum of three and zero?
Expectated:
the answer is three
Output:

Enter a question:

The answer is three

3

Question:

What do I get if I add seventy-five  and forty-nine together?
Expectated:
the answer is not a number I can put into words (124)
Output:

Enter a question:
```



```
The answer is not a number I can put into words (118)

4

Question:

What is forty-seven minus twenty-seven?
Expectated:
the answer is twenty
Output:

Enter a question:

The answer is twenty

5

Question:

ninety less eighty is what?
Expectated:
the answer is ten
Output:

Enter a question:

The answer is ten

6

Question:

What is twelve times six?
Expectated:
the answer is seventy-two
Output:

Enter a question:

The answer is seventy-two

7

Question:

What do I get when I multiply fifteen and zero?
Expectated:
the answer is zero
Output:
```

-1 (execution) wrong result

```
Enter a question:

The answer is not a number I can put into words (0)

8

Question:

What is seventy-five divided by fifteen?
Expectated:
the answer is five
Output:

Enter a question:
```

```
The answer is five


9

Question:

What is seventy-five divided by thirteen?
Expectated:
the answer is five
Output:

Enter a question:

The answer is five


10

Question:

This is an error sentence?
Expectated:
I'm sorry, I don't understand the question. Please return and rephrase your ques
tion.
Output:

Enter a question:

I'm sorry, I don't understand the question. Please rerun and rephrase your quest
ion.


11

Question:

Tell me the sum of 3 and 27?
Expectated:
I'm sorry, I don't understand the question. Please return and rephrase your ques
tion.
Output:

Enter a question:

I'm sorry, I don't understand the question. Please rerun and rephrase your quest
ion.


##########################################
Another Test


Running TestP3.java


============== Questions with standard templates ================
Question:
 What is fifty-five plus sixty-seven?
Expected:
 the answer is not a number I can put into words (122)
Got:
 the answer is not a number I can put into words (122)

Question:
 Tell me the sum of tres and sixty?
Expected:
 Question not understood
```



```
Got:
 Question not understood

Question:
 Tell me the sum of twenty-five and fifty-trois?
Expected:
 Question not understood
Got:
 the answer is seventy-four

Question:
 What do I get if I add twenty-five and sixty together?
Expected:
 the answer is eighty-five
Got:
 the answer is eighty-five

Question:
 What is fifty-five minus twenty-one?
Expected:
 the answer is thirty-four
Got:
 the answer is thirty-four

Question:
 What is fifty-five minus sixty-six?
Expected:
 the answer is not a number I can put into words (-11)
Got:
 the answer is not a number I can put into words (-11)

Question:
 Fifty-five less twenty-one is what?
Expected:
 the answer is thirty-four
Got:
 the answer is thirty-four

Question:
 What is twenty-five times three?
Expected:
 the answer is seventy-five
Got:
 the answer is seventy-five

Question:
 What is twenty-one times twenty-two?
Expected:
 the answer is not a number I can put into words (462)
Got:
 the answer is not a number I can put into words (462)

Question:
 What is sixty-four divided by twelve?
Expected:
 the answer is five
Got:
 the answer is five

Question:
 What is twenty-four divided by thirty-three?
Expected:
 the answer is zero
Got:
 the answer is not a number I can put into words (0)
============================================================
====================== Words to Numbers ======================
Input:
 zero  Expected:
```

-0.25 (execution) wrong result

-0.25 (execution) wrong result

```
 0
Got:
 0

Input:
 fifteen  Expected:
 15
Got:
 15

Input:
 twenty  Expected:
 20
Got:
 20

Input:
 forty  Expected:
 40
Got:
 40

Input:
 forty-two  Expected:
 42
Got:
 42

Input:
 ninety-nine  Expected:
 99
Got:
 99

Input:
 tres  Expected:
 -1
Got:
 -1

Input:
 zippity-nine  Expected:
 -1
Got:
 8

Input:
 fifty-trois  Expected:
 -1
Got:
 49
================================================================

====================== Numbers to Words ======================
Input:
 0  Expected:
 zero
Got:
 not a number I can put into words (0)

Input:
 7  Expected:
 seven
Got:
 seven

Input:
 20  Expected:
 twenty
```

-0.5 (execution) wrong result

-0.5 (execution) wrong result

-0.5 (execution) wrong result


```
Got:
 twenty

Input:
 70  Expected:
 seventy
Got:
 seventy

Input:
 88  Expected:
 eighty-eight
Got:
 eighty-eight

Input:
 99  Expected:
 ninety-nine
Got:
 ninety-nine

Input:
 -12  Expected:
 not a number I can put into words (-12)
Got:
 not a number I can put into words (-12)

Input:
 1000  Expected:
 not a number I can put into words (1000)
Got:
 not a number I can put into words (1000)
================================================================
======================== Expressions =========================
Question:
 AdditionExpression("nine", "twenty-two")
Expected:
 The answer is 31
Got:
 The answer is 31

Question:
 SubtractionExpression("four", "three")
Expected:
 The answer is 1
Got:
 The answer is 1

Question:
 MultiplicationExpression("four", "nineteen")
Expected:
 The answer is 76
Got:
 The answer is 76

Question:
 DivisionExpression("seventy-two", "two")
Expected:
 The answer is 36
Got:
 The answer is 36
================================================================
==================== Custom Templates ========================
Question:
 + twenty-five & three *
Expected:
 The answer is twenty-eight
Got:
 The answer is twenty-eight
```

```
Question:
 Test Template twenty-five for three subtraction
Expected:
 The answer is twenty-two
Got:
 The answer is twenty-two

Question:
 Multiplication   twenty-five   test   three    template
Expected:
 The answer is seventy-five
Got:
 Question not understood

Question:
 Division    twenty-five template five test
Expected:
 The answer is five
Got:
 The answer is not a number I can put into words (0)
================================================================
```

-0.5 (execution) wrong result

-0.5 (execution) wrong result