Name: Christopher Chan

Student Number: 20143087

Class: HDSDEV_SEP

Module: Software Development

Find Computer Words Game Report

Input Output and Processing

Input:

- Ask the players how many rounds they would like to play
- Read in the player1 and player2's inputted words

Processing:

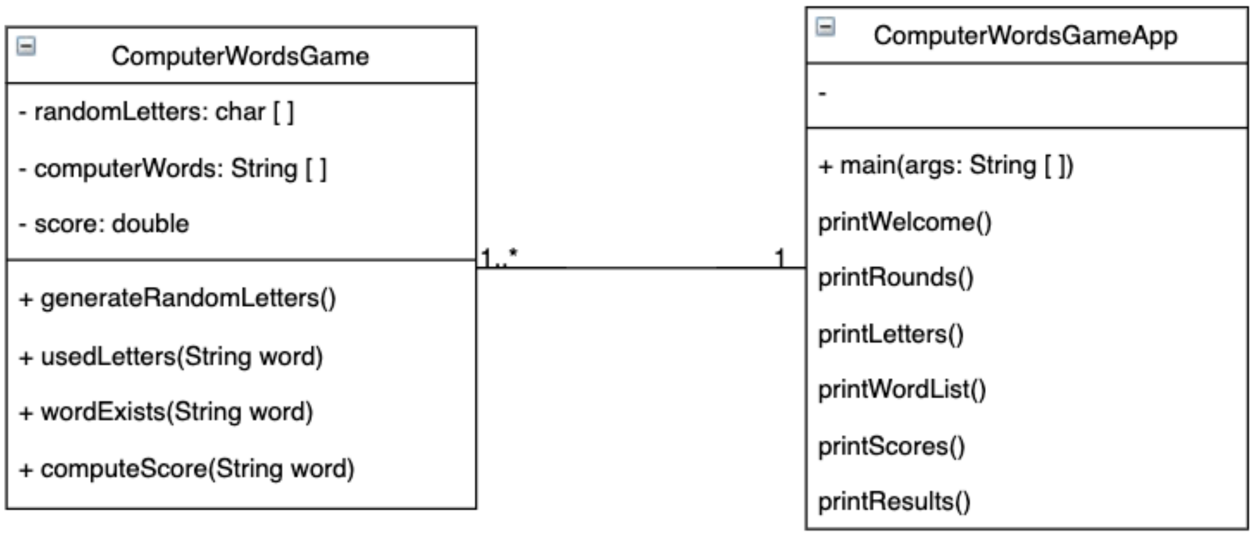
- Calculate the amount of points each player receives from their words

Output

- Display the points received by each player
- Display the points per round
- Display the total points each player receives and the game winner

Class Diagram

In order to implement the specification of the game as simply as possibly. I have two classes, the ComputerWordsGame class and the ComputerWordsGameApp class, that contains the main method. I excluded the constructors and getter methods from the ComputerWordsGame class because it not really necessary to show them for the class diagram.

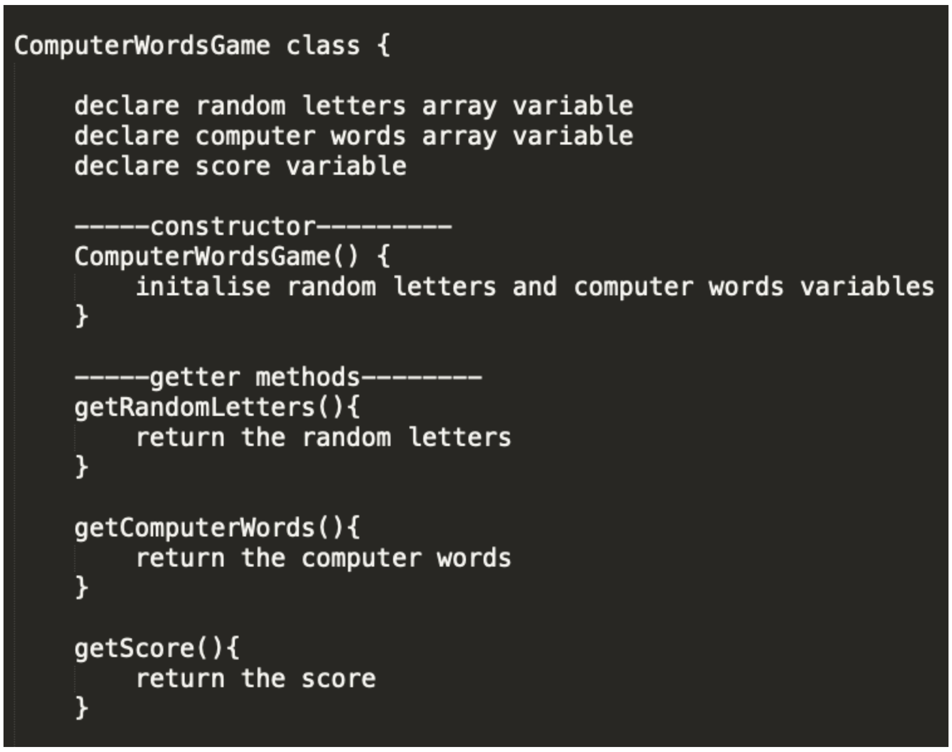


Class Diagram for Find Computer Words Game

Design and implementation

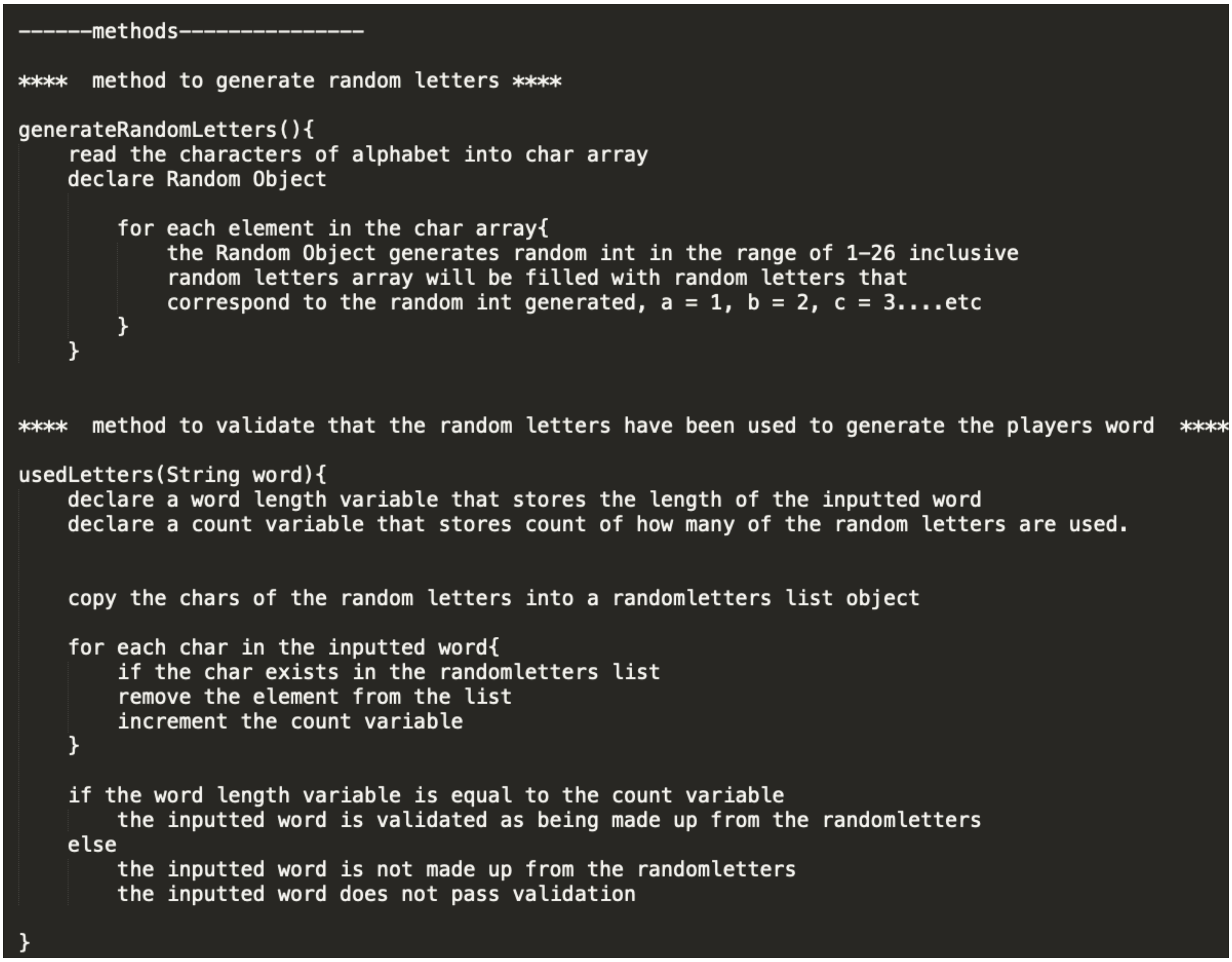
In order to make the instantiable ComputerWordsGame class as simple and re-usable as possible. I decide to have one constructor that initialises the data, three getter methods to access the data, and four methods that take care of the validation and the processing of the game. I believe the data and the methods defined are decoupled to the extent that in future if the game requires any tweaks or changes in rules, it is easy to write new methods and functions to expand this class's capabilities.

Below is the pseudocode I used to structure the logic and map out how I was going to implement the classes.



```
ComputerWordsGame class {

    declare random letters array variable
    declare computer words array variable
    declare score variable

    -----constructor---------
    ComputerWordsGame() {
        initalise random letters and computer words variables
    }

    -----getter methods--------
    getRandomLetters(){
        return the random letters
    }

    getComputerWords(){
        return the computer words
    }

    getScore(){
        return the score
    }
```

ComputerWordsGame class Pseudocode fig1

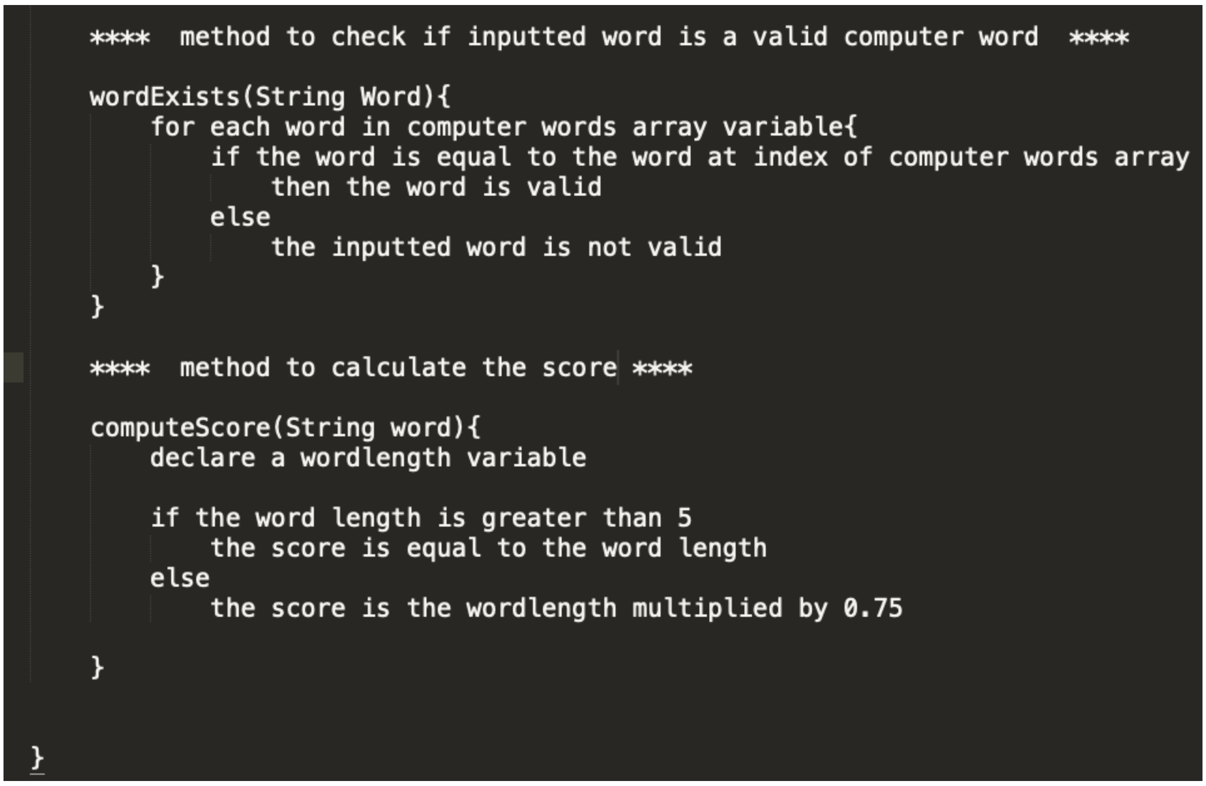


```
------methods---------------

****  method to generate random letters ****

generateRandomLetters(){
    read the characters of alphabet into char array
    declare Random Object

        for each element in the char array{
            the Random Object generates random int in the range of 1-26 inclusive
            random letters array will be filled with random letters that
            correspond to the random int generated, a = 1, b = 2, c = 3....etc
        }
    }


****  method to validate that the random letters have been used to generate the players word  ****

usedLetters(String word){
    declare a word length variable that stores the length of the inputted word
    declare a count variable that stores count of how many of the random letters are used.


    copy the chars of the random letters into a randomletters list object

    for each char in the inputted word{
        if the char exists in the randomletters list
        remove the element from the list
        increment the count variable
    }

    if the word length variable is equal to the count variable
        the inputted word is validated as being made up from the randomletters
    else
        the inputted word is not made up from the randomletters
        the inputted word does not pass validation

}
```

ComputerWordsGame class Pseudocode fig2

ComputerWordsGame Class

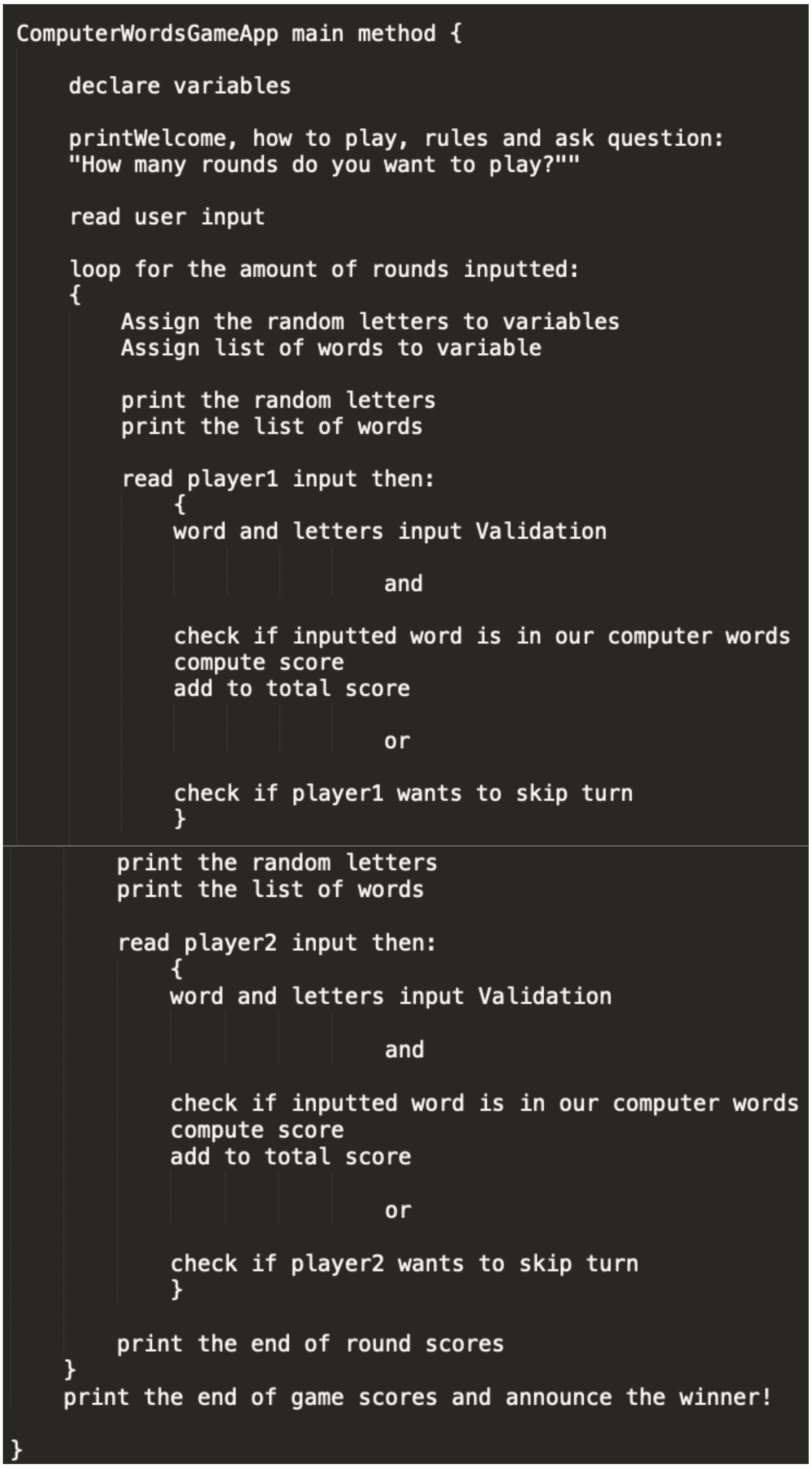
In *fig2* I used a arraylist object to do the validation in the usedLetters method due to the mutable characteristics that lists have over arrays *ref1*.



```
    ****  method to check if inputted word is a valid computer word  ****

    wordExists(String Word){
        for each word in computer words array variable{
            if the word is equal to the word at index of computer words array
                then the word is valid
            else
                the inputted word is not valid
        }
    }

    ****  method to calculate the score ****

    computeScore(String word){
        declare a wordlength variable

        if the word length is greater than 5
            the score is equal to the word length
        else
            the score is the wordlength multiplied by 0.75

    }


}
```

ComputerWordsGame class Pseudocode fig3

ComputerWordsGameApp Class

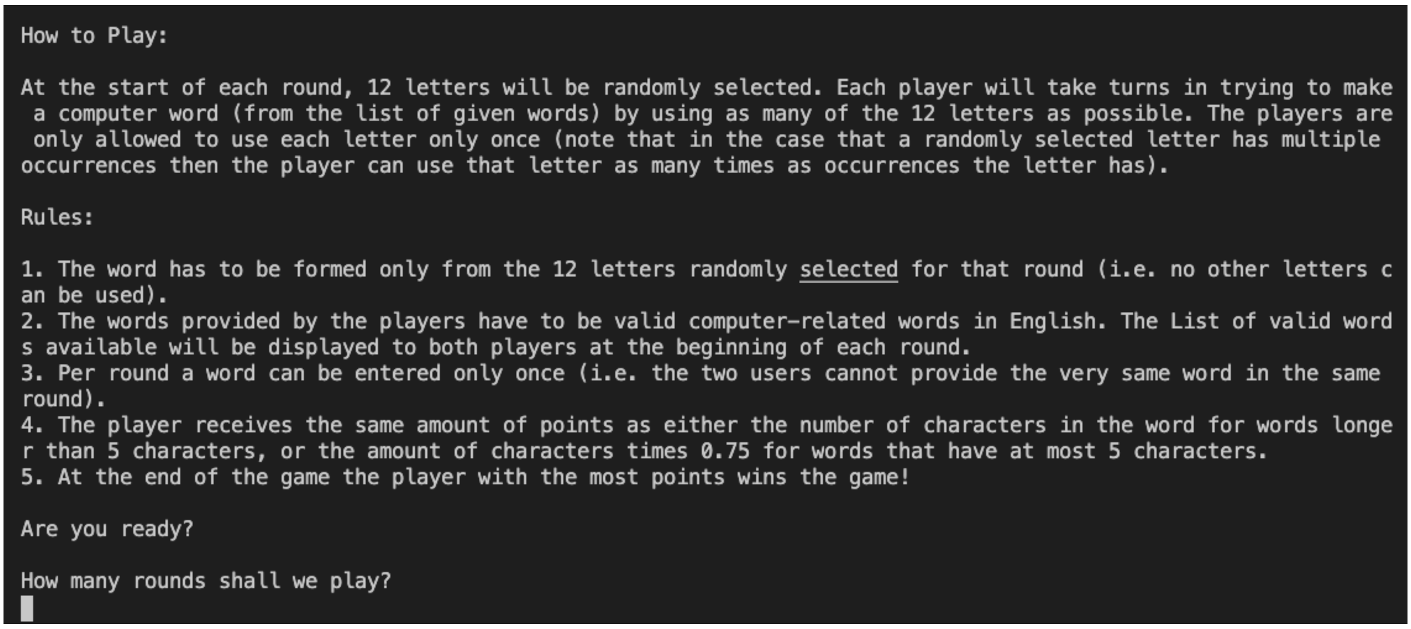


```
ComputerWordsGameApp main method {

    declare variables

    printWelcome, how to play, rules and ask question:
    "How many rounds do you want to play?""

    read user input

    loop for the amount of rounds inputted:
    {
        Assign the random letters to variables
        Assign list of words to variable

        print the random letters
        print the list of words

        read player1 input then:
            {
            word and letters input Validation

                            and

            check if inputted word is in our computer words
            compute score
            add to total score

                            or

            check if player1 wants to skip turn
            }
        print the random letters
        print the list of words

        read player2 input then:
            {
            word and letters input Validation

                            and

            check if inputted word is in our computer words
            compute score
            add to total score

                            or

            check if player2 wants to skip turn
            }

        print the end of round scores
    }
    print the end of game scores and announce the winner!

}
```

ComputerWordsGameApp class Pseudocode fig1

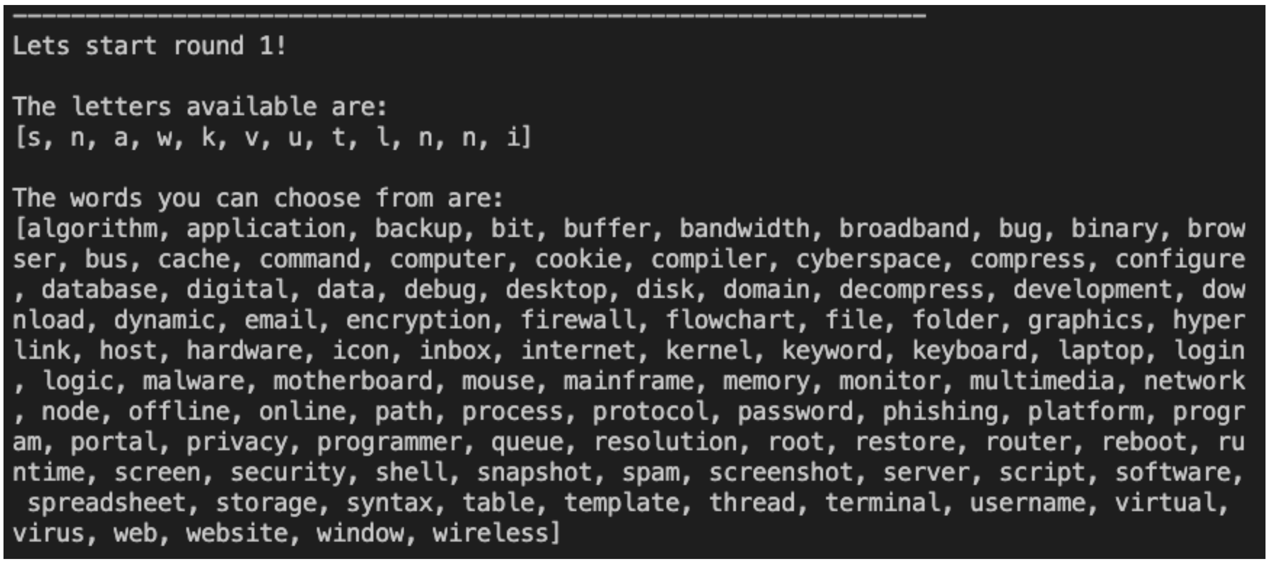
Playing the game

When the program is run the players are greeted with the welcome screen. The players will be informed on how to play the game as well as the rules. The players will then have to select how many rounds they would like to play.



```
How to Play:

At the start of each round, 12 letters will be randomly selected. Each player will take turns in trying to make
 a computer word (from the list of given words) by using as many of the 12 letters as possible. The players are
 only allowed to use each letter only once (note that in the case that a randomly selected letter has multiple
occurrences then the player can use that letter as many times as occurrences the letter has).

Rules:

1. The word has to be formed only from the 12 letters randomly selected for that round (i.e. no other letters c
an be used).
2. The words provided by the players have to be valid computer-related words in English. The List of valid word
s available will be displayed to both players at the beginning of each round.
3. Per round a word can be entered only once (i.e. the two users cannot provide the very same word in the same
round).
4. The player receives the same amount of points as either the number of characters in the word for words longe
r than 5 characters, or the amount of characters times 0.75 for words that have at most 5 characters.
5. At the end of the game the player with the most points wins the game!

Are you ready?

How many rounds shall we play?
```

The Welcome Screen

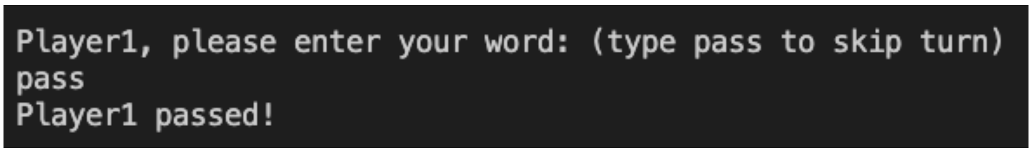


```
-------------------------------------------------------------
Lets start round 1!

The letters available are:
[s, n, a, w, k, v, u, t, l, n, n, i]

The words you can choose from are:
[algorithm, application, backup, bit, buffer, bandwidth, broadband, bug, binary, brow
ser, bus, cache, command, computer, cookie, compiler, cyberspace, compress, configure
, database, digital, data, debug, desktop, disk, domain, decompress, development, dow
nload, dynamic, email, encryption, firewall, flowchart, file, folder, graphics, hyper
link, host, hardware, icon, inbox, internet, kernel, keyword, keyboard, laptop, login
, logic, malware, motherboard, mouse, mainframe, memory, monitor, multimedia, network
, node, offline, online, path, process, protocol, password, phishing, platform, progr
am, portal, privacy, programmer, queue, resolution, root, restore, router, reboot, ru
ntime, screen, security, shell, snapshot, spam, screenshot, server, script, software,
 spreadsheet, storage, syntax, table, template, thread, terminal, username, virtual,
virus, web, website, window, wireless]
```

Round 1

Player1 will start the game, the random letters and the computer words will be displayed to the player.

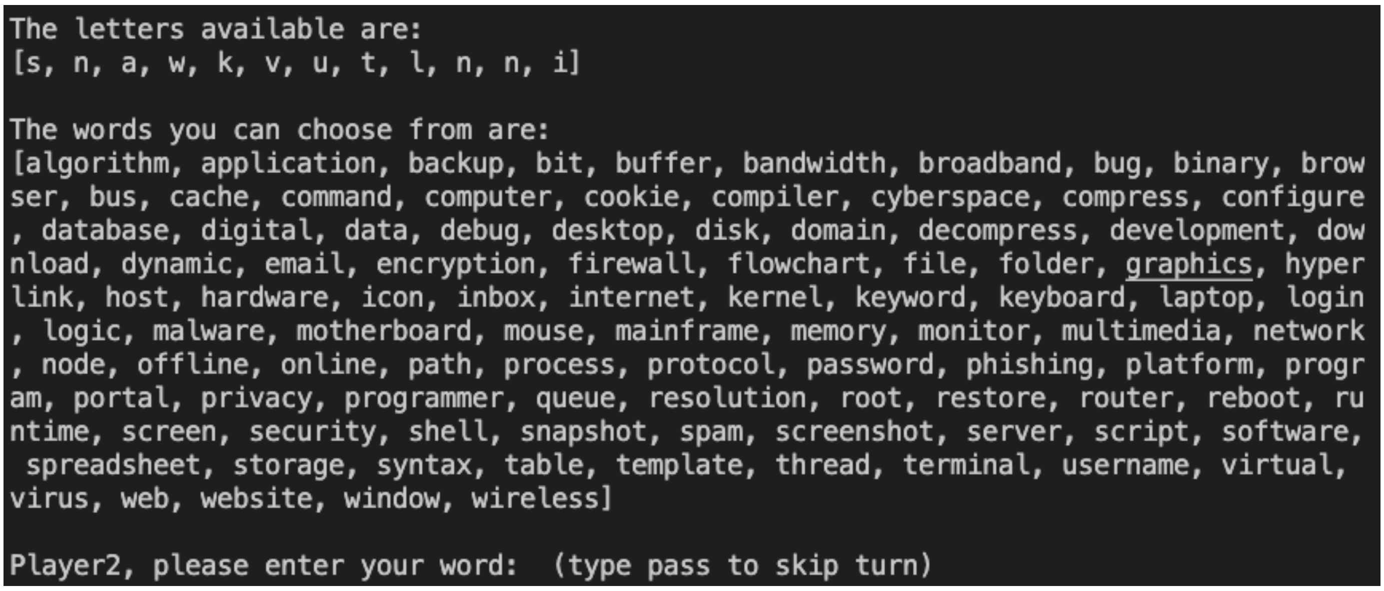
If player1 can't make a word with the random letters they also have an option to skip their turn by typing "pass".



```
Player1, please enter your word: (type pass to skip turn)
pass
Player1 passed!
```

Player1 has passed

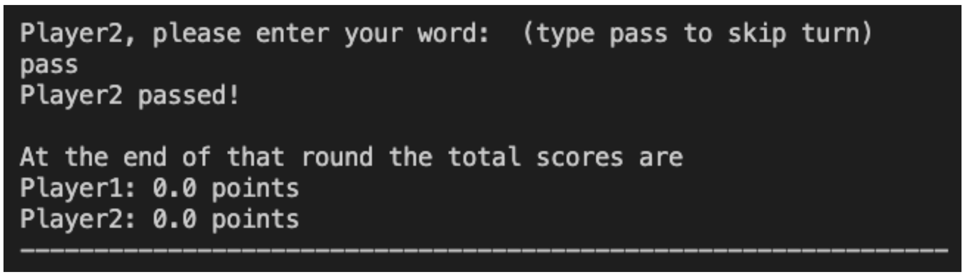
Next is it player2's turn.



```
The letters available are:
[s, n, a, w, k, v, u, t, l, n, n, i]

The words you can choose from are:
[algorithm, application, backup, bit, buffer, bandwidth, broadband, bug, binary, brow
ser, bus, cache, command, computer, cookie, compiler, cyberspace, compress, configure
, database, digital, data, debug, desktop, disk, domain, decompress, development, dow
nload, dynamic, email, encryption, firewall, flowchart, file, folder, graphics, hyper
link, host, hardware, icon, inbox, internet, kernel, keyword, keyboard, laptop, login
, logic, malware, motherboard, mouse, mainframe, memory, monitor, multimedia, network
, node, offline, online, path, process, protocol, password, phishing, platform, progr
am, portal, privacy, programmer, queue, resolution, root, restore, router, reboot, ru
ntime, screen, security, shell, snapshot, spam, screenshot, server, script, software,
 spreadsheet, storage, syntax, table, template, thread, terminal, username, virtual,
virus, web, website, window, wireless]

Player2, please enter your word:  (type pass to skip turn)
```

Player2's turn

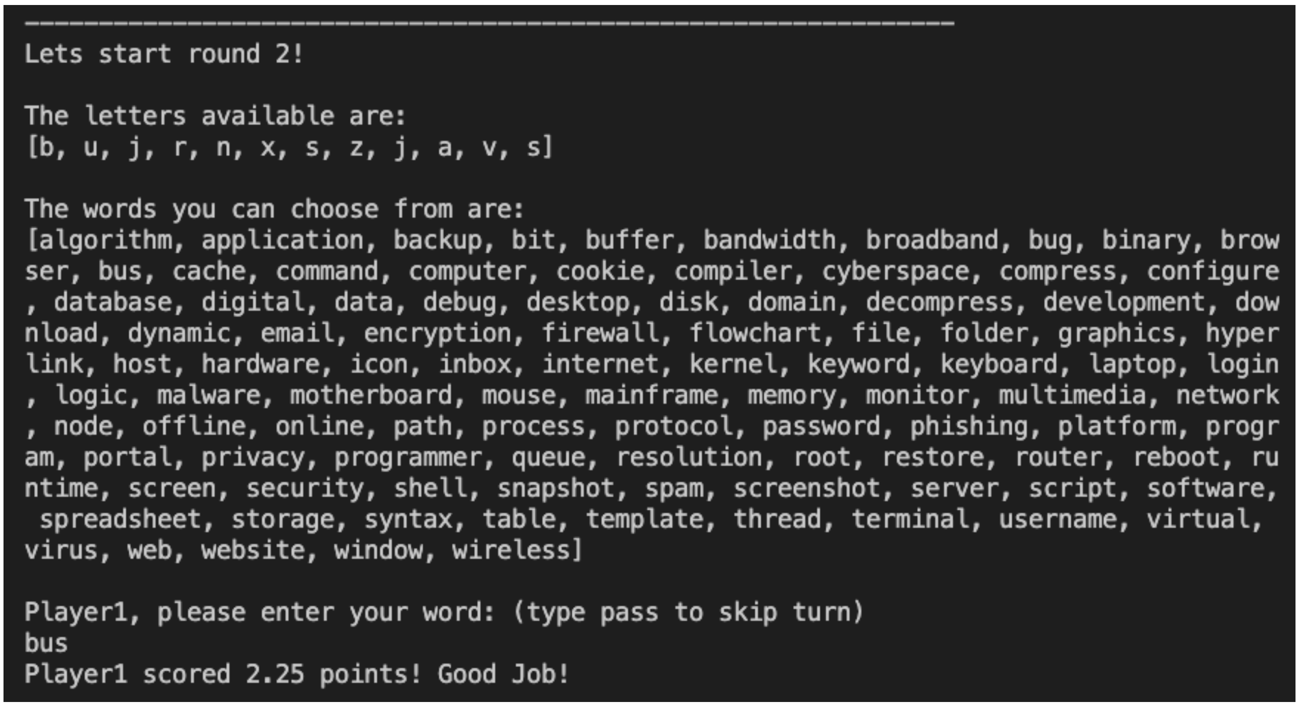
If player2 passes too, then the round ends and the scores are shown to both players.



```
Player2, please enter your word:  (type pass to skip turn)
pass
Player2 passed!

At the end of that round the total scores are
Player1: 0.0 points
Player2: 0.0 points
---------------------------------------------------------------
```

The round ends

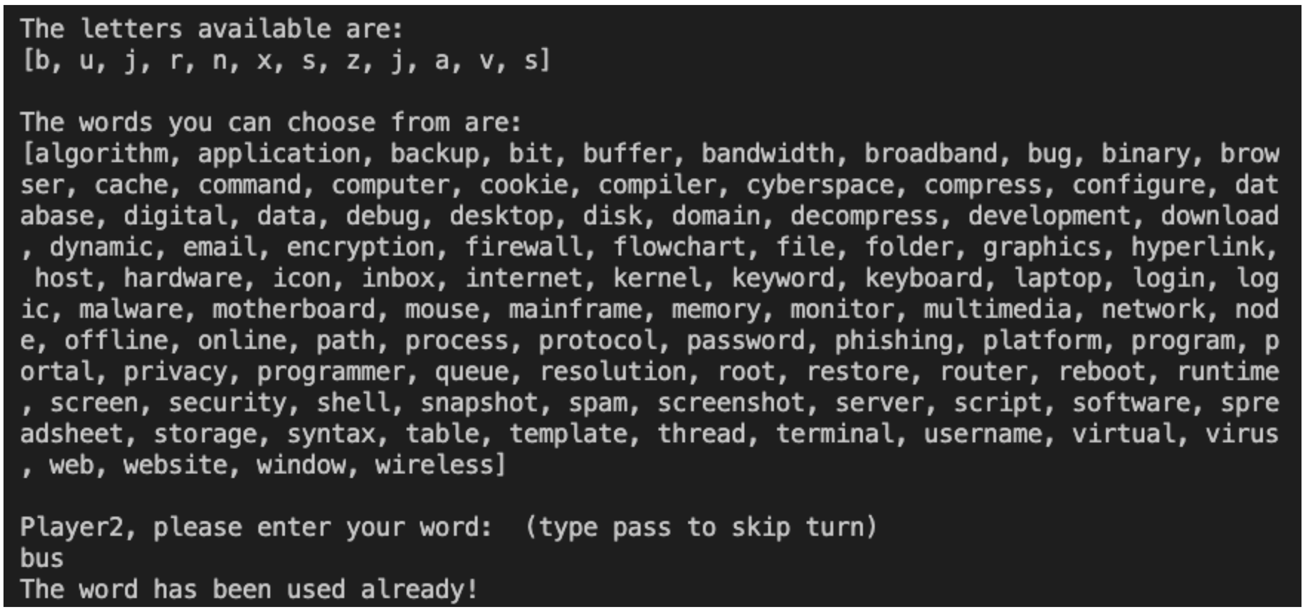
In round 2 its player1's turn. Player1 manages to make a word with 3 letters and scores 2.25 points.



```
----------------------------------------------------------------
Lets start round 2!

The letters available are:
[b, u, j, r, n, x, s, z, j, a, v, s]

The words you can choose from are:
[algorithm, application, backup, bit, buffer, bandwidth, broadband, bug, binary, brow
ser, bus, cache, command, computer, cookie, compiler, cyberspace, compress, configure
, database, digital, data, debug, desktop, disk, domain, decompress, development, dow
nload, dynamic, email, encryption, firewall, flowchart, file, folder, graphics, hyper
link, host, hardware, icon, inbox, internet, kernel, keyword, keyboard, laptop, login
, logic, malware, motherboard, mouse, mainframe, memory, monitor, multimedia, network
, node, offline, online, path, process, protocol, password, phishing, platform, progr
am, portal, privacy, programmer, queue, resolution, root, restore, router, reboot, ru
ntime, screen, security, shell, snapshot, spam, screenshot, server, script, software,
 spreadsheet, storage, syntax, table, template, thread, terminal, username, virtual,
virus, web, website, window, wireless]

Player1, please enter your word: (type pass to skip turn)
bus
Player1 scored 2.25 points! Good Job!
```

Player1 makes a word

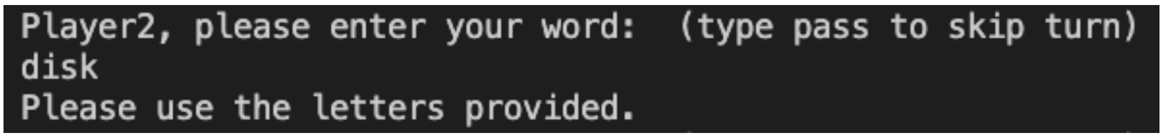
The word is now removed from the list of choosable computer words for player2. If player2 tries to use the same word, they will get a message saying the word has been used already.



```
The letters available are:
[b, u, j, r, n, x, s, z, j, a, v, s]

The words you can choose from are:
[algorithm, application, backup, bit, buffer, bandwidth, broadband, bug, binary, brow
ser, cache, command, computer, cookie, compiler, cyberspace, compress, configure, dat
abase, digital, data, debug, desktop, disk, domain, decompress, development, download
, dynamic, email, encryption, firewall, flowchart, file, folder, graphics, hyperlink,
 host, hardware, icon, inbox, internet, kernel, keyword, keyboard, laptop, login, log
ic, malware, motherboard, mouse, mainframe, memory, monitor, multimedia, network, nod
e, offline, online, path, process, protocol, password, phishing, platform, program, p
ortal, privacy, programmer, queue, resolution, root, restore, router, reboot, runtime
, screen, security, shell, snapshot, spam, screenshot, server, script, software, spre
adsheet, storage, syntax, table, template, thread, terminal, username, virtual, virus
, web, website, window, wireless]

Player2, please enter your word:  (type pass to skip turn)
bus
The word has been used already!
```

Player2 cannot input the same word

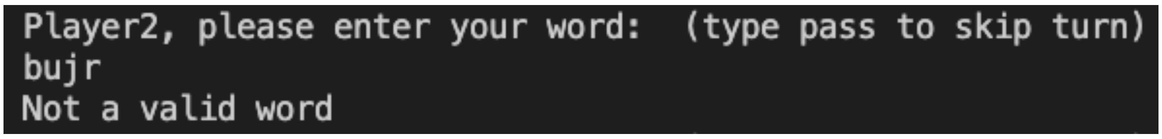
If player 2 tries to choose a word from the list without using the random letters. They will be asked to use the letters provided.



```
Player2, please enter your word:  (type pass to skip turn)
disk
Please use the letters provided.
```

Player2 has to use the random letters only

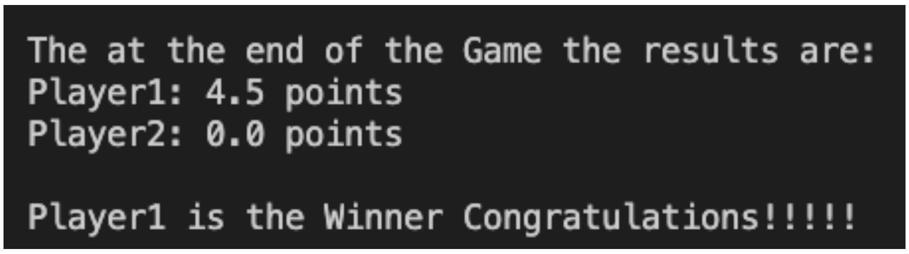
If player2 uses the letters provided but it is not a valid computer word. They will be informed it is not a valid word.



```
Player2, please enter your word:  (type pass to skip turn)
bujr
Not a valid word
```

Player2 entered and invalid word

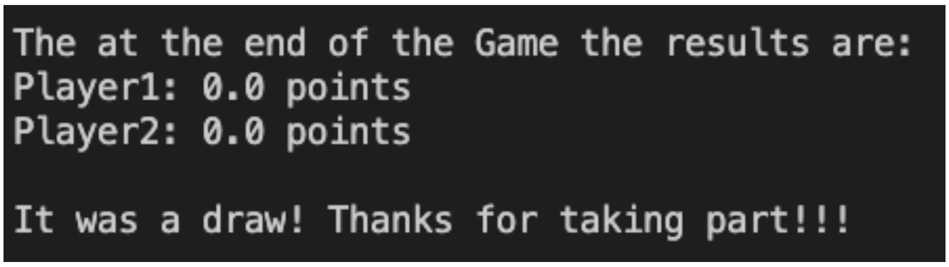
At the end of the last round. The final points tally will be shown and information about who has won the game will be displayed.



```
The at the end of the Game the results are:
Player1: 4.5 points
Player2: 0.0 points

Player1 is the Winner Congratulations!!!!!
```

Player1 won the game

If the game ends in a draw that will be displayed too.



```
The at the end of the Game the results are:
Player1: 0.0 points
Player2: 0.0 points

It was a draw! Thanks for taking part!!!
```

The game ended in a draw

References:

1. Array List: https://www.w3schools.com/java/java_arraylist.asp

Source Code:

```
import java.util.List;
import java.util.ArrayList;
import java.util.Random;

public class ComputerWordsGame {
    private char[] randomLetters;
    private String[] computerWords;
    private double score;

    // ---------constructor--------------
    public ComputerWordsGame() {
        randomLetters = new char[12];
        computerWords = new String[] { "algorithm",
"application", "backup", "bit", "buffer", "bandwidth", "broad-
band",
                "bug", "binary", "browser", "bus", "cache", "com-
mand", "computer", "cookie", "compiler", "cyberspace",
                "compress", "configure", "database", "digital",
"data", "debug", "desktop", "disk", "domain",
                "decompress", "development", "download", "dynam-
ic", "email", "encryption", "firewall", "flowchart",
                "file", "folder", "graphics", "hyperlink",
"host", "hardware", "icon", "inbox", "internet", "kernel",
                "keyword", "keyboard", "laptop", "login",
"logic", "malware", "motherboard", "mouse", "mainframe",
                "memory", "monitor", "multimedia", "network",
"node", "offline", "online", "path", "process",
                "protocol", "password", "phishing", "platform",
"program", "portal", "privacy", "programmer", "queue",
                "resolution", "root", "restore", "router", "re-
boot", "runtime", "screen", "security", "shell",
                "snapshot", "spam", "screenshot", "server",
"script", "software", "spreadsheet", "storage", "syntax",
                "table", "template", "thread", "terminal", "user-
name", "virtual", "virus", "web", "website", "window",
                "wireless" };
    }

    // ----------getters -------------
    public char[] getRandomLetters() {
        return randomLetters;
    }

    public String[] getComputerWords() {
        return computerWords;
    }

    public double getScore() {
        return score;
    }

    // --------methods ------------------
    public void generateRandomLetters() {
        String alphabetString = "abcdefghijklmnopqrstuvwxyz";
        char[] alphabet = alphabetString.toCharArray();
        Random random = new Random();
        int randomNumber;

        for (int i = 0; i < randomLetters.length; i++) {
            randomNumber = random.nextInt(26);
            randomLetters[i] = alphabet[randomNumber];
        }
    }

    // A method to find out that only the letters given are used
the make the entire
    // word
    public boolean usedLetters(String word) {
        List<Character> randomLettersList = new ArrayList<>();
        int count = 0;
        int wordLength = 0;

        // getting the length of the inputted word
        for (int i = 0; i < word.length(); i++) {
            if (word.charAt(i) != ' ') {
                wordLength++;
            }
        }

        // copying the array of chars to the arraylist obj ran-
domLettersList
        for (int i = 0; i < randomLetters.length; i++) {
            randomLettersList.add(randomLetters[i]);
        }

        // for each character of the word. each occurance is re-
moved from the
        // randomLettersList and the count is incremented
        for (int i = 0; i < wordLength; i++) {
            if (randomLettersList.contains(word.charAt(i))) {
                int index =
randomLettersList.indexOf(word.charAt(i));
                randomLettersList.remove(index);
                count++;
            }
        }
        // if the wordLength is equal to the count. The player
used valid letters from
        // the list
        if (wordLength == count) {
            return true;
        }
        return false;
    }

    public boolean wordExists(String word) {
        // lets do it the olde brute force iterative search way
        for (int i = 0; i < computerWords.length; i++) {
            if (computerWords[i].equals(word)) {
                return true;
            }
        }
        return false;
```

```
    }

    public void computeScore(String word) {
        int wordLength = 0;
        for (int i = 0; i < word.length(); i++) {
            if (word.charAt(i) != ' ') {
                wordLength++;
            }
        }
        if (wordLength > 5) {
            score = wordLength;
        } else {
            score = ((double) wordLength) * 0.75;
        }
    }
}

import java.util.List;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Scanner;

public class ComputerWordsGameApp {
    public static void main(String args[]) {

        Scanner input = new Scanner(System.in);
        ComputerWordsGame game = new ComputerWordsGame();
        List<String> wordsList;
        char[] letters;
        int player, round, roundCounter = 1;
        double player1Score = 0, player2Score = 0;

        printWelcome();
        round = input.nextInt();

        for (int i = 0; i < round; i++) {

            printRounds(roundCounter);
            // add all the words into our arrayList
            wordsList = new ArrayList<String>();
            Collections.addAll(wordsList, 
game.getComputerWords());

            game.generateRandomLetters();
            letters = game.getRandomLetters();

            printLetters(letters);
            printWordList(wordsList);

            String inputWord, wordCheck = "";

            player = 1;
            while (true) {
                System.out.println("Player" + player + ", please 
enter your word: (type pass to skip turn)");
                inputWord = input.next().toLowerCase();
                if (game.wordExists(inputWord) && game.usedLet-
ters(inputWord)) {
                    wordCheck = inputWord;
                    game.computeScore(inputWord);
                    System.out.println("Player" + player + " 
scored " + game.getScore() + " points! Good Job!");
                    player1Score = player1Score + 
game.getScore();

                    // remove the element from the arraylist
                    int index = wordsList.indexOf(inputWord);
                    wordsList.remove(index);
                    break;

                } else if (inputWord.equals("pass")) {
                    System.out.println("Player" + player + " 
passed!");
                    break;
                } else if (!game.usedLetters(inputWord)) {
                    System.out.println("Please use the letters 
provided.");
```

```
                } else {
                    System.out.println("Not a valid word");
                }
            }

            printLetters(letters);
            printWordList(wordsList);

            player = 2;
            while (true) {
                System.out.println("Player" + player + ", please 
enter your word:  (type pass to skip turn)");
                inputWord = input.next().toLowerCase();
                if (!wordCheck.equals(inputWord)) {
                    if (game.wordExists(inputWord) && game.used-
Letters(inputWord)) {
                        game.computeScore(inputWord);
                        System.out.println("Player" + player + " 
scored " + game.getScore() + " points! Good Job!");
                        player2Score = player2Score + game.get-
Score();

                        break;
                    } else if (inputWord.equals("pass")) {
                        System.out.println("Player" + player + " 
passed!");

                        break;
                    } else if (!game.usedLetters(inputWord)) {
                        System.out.println("Please use the let-
ters provided.");
                    } else {
                        System.out.println("Not a valid word");
                    }
                } else {
                    System.out.println("The word has been used 
already!");
                }
            }
            printScores(player1Score, player2Score);
            roundCounter++;
```

```
        }
        printResults(player1Score, player2Score);
        input.close();
    }

    static void printWelcome() {
        System.out.println();
        System.out.println("Welcome to the Find Computer Words 
Game! ");
        System.out.println();
        System.out.println("How to Play: ");
        System.out.println();
        System.out.println(
                "At the start of each round, 12 letters will be 
randomly selected. Each player will take turns in trying to make 
a computer word (from the list of given words) by using as many 
of the 12 letters as possible. The players are only allowed to 
use each letter only once (note that in the case that a randomly 
selected letter has multiple occurrences then the player can use 
that letter as many times as occurrences the letter has).");
        System.out.println();
        System.out.println("Rules:");
        System.out.println();
        System.out.println(
                "1. The word has to be formed only from the 12 
letters randomly selected for that round (i.e. no other letters 
can be used).");
        System.out.println(
                "2. The words provided by the players have to be 
valid computer-related words in English. The List of valid words 
available will be displayed to both players at the beginning of 
each round.");
        System.out.println(
                "3. Per round a word can be entered only once 
(i.e. the two users cannot provide the very same word in the same 
round).");
        System.out.println(
                "4. The player receives the same amount of points 
as either the number of characters in the word for words longer 
```

```
than 5 characters, or the amount of characters times 0.75 for
words that have at most 5 characters.");
        System.out.println("5. At the end of the game the player
with the most points wins the game!");
        System.out.println();
        System.out.println("Are you ready?");
        System.out.println();
        System.out.println("How many rounds shall we play?");

    }

    static void printRounds(int roundCounter) {
        System.out.println("-------------------------------------
--------------------------");
        System.out.println("Lets start round " + roundCounter +
"!");
    }

    static void printLetters(char[] letters) {
        System.out.println();
        System.out.println("The letters available are: ");
        System.out.print("[");
        for (int i = 0; i < letters.length; i++) {
            System.out.print(letters[i]);
            if (i != 11) {
                System.out.print(", ");
            }
        }
        System.out.println("]");
        System.out.println();
    }

    static void printWordList(List<String> wordsList) {
        System.out.println("The words you can choose from are:
");
        System.out.println(wordsList);
        System.out.println();
    }
    static void printScores(double player1Score, double player2S-
core) {
        System.out.println();
        System.out.println("At the end of that round the total
scores are");
        System.out.println("Player1: " + player1Score + "
points");
        System.out.println("Player2: " + player2Score + "
points");
        System.out.println("-------------------------------------
--------------------------");
        System.out.println();
    }

    static void printResults(double player1Score, double play-
er2Score) {
        System.out.println("The at the end of the Game the re-
sults are: ");
        System.out.println("Player1: " + player1Score + "
points");
        System.out.println("Player2: " + player2Score + "
points");
        System.out.println();

        if (player1Score > player2Score) {
            System.out.println("Player1 is the Winner Congratula-
tions!!!!!");
        } else if (player1Score < player2Score) {
            System.out.println("Player2 is the Winner Congratula-
tions!!!!!");
        } else {
            System.out.println("It was a draw! Thanks for taking
part!!!");
        }
        System.out.println();
    }

}
```