Assignment Q1

To be completed in week 1

Create a new Microsoft Word document and call your document as YourFirstNameLastName-

A1Q1.docx.

Open the house project from chapter 1 of the reference book projects.

Open the terminal window and record method calls.

Create a picture with at least six (6) objects (circle, square, triangle, and person), recording

all method calls.

Take a screenshot of your picture and add it to the word document you created earlier. Then

copy the list of method calls needed to create the picture and paste them in the word

document after the screenshot.

Add your name and student ID in the footer of the word document, as well as "PROG2007

Assignment 1 Q1".

Ans.=>

public class AssignmentTest {

public static void main(String[] args){

Assignment assignment1 = new Assignment("abc",20,15);

assignment1.calculateGrade();

System.out.println("Assignment 1 results : \nStudent Mark : " + assignment1.getStudentMark() + "\nGrade : " + assignment1.getGrade());

}

}

--------------------------------------------------------------------------------------------------------------------------------

Assignment Q2

To be completed in week 2

Create a new BlueJ project called YourFirstNameLastName-A1Q2.

Create a class called Employee.

Make sure you write a description of your new Class in the comments, with your name as

author and date as the last date you worked on this exercise.

Add definitions for the following fields:

• A name field of type String

• An employeeId field of type int

• A wage field of type double

• A fullTime field of type boolean

Write a constructor for your Employee class that takes four parameters - the first of type

String called myName, the second of type int called myEmployeeId, the third of type double

called myWage, and the fourth of type boolean called isFullTime. Set the initial values of the

corresponding fields using the constructor.

Write an accessor method called getName that returns the value of the name field.

3

Write a mutator method called setEmployeeId that takes a single parameter of type int and

sets the value of the employeeId field.

Work out what other accessor and mutator methods would be useful for this Class and add

them. You should be able to get and set all fields in the Class.

Write a method called printDetails, which prints out all the details of an Employee object. You

must take into account the fullTime status and print a line saying either that the employee is

fulltime or the employee is not fulltime.

For example, if:

• The name field holds the value "John Smith"

• The employeeId field holds the value 123456

• The wage field holds the value 25.40

• The fullTime field holds the value false

Then the printDetails method would print out the following:

The name of the employee is John Smith. The employee id is 123456. The wage of the

employee is $25.40 per hour. The employee is fulltime.

If the fulltime field holds the value false, then the printDetails method would print out the

following:

The name of the employee is John Smith. The employee id is 123456. The wage of the

employee is $25.40 per hour. The employee is not fulltime.

Please Note: In the above examples, the name, employeeId, wage and whether the employee

is fulltime or not (in blue) will change based on the values the fields hold. However, you must

print the remainder of the statements exactly as in the above examples.

Ans.=>

class Worker {

private String name;

private int workerId;

private double wage;

private boolean fullTime;

Worker(String name, int workerId, double wage, boolean fullTime){

this.name = name;

this.workerId = workerId;

this.wage = wage;

this.fullTime=fullTime;

}

public String getName(){

return this.name;

}

public void setWorkerId(int id){

this.workerId = id;

}

public int getWorkerId(){

return this.workerId;

}

public double getWage(){

return this.wage;

}

public boolean getFullTime(){

return this.fullTime;

}

public void setName(String n){

this.name = n;

}

public void setWage(double w){

this.wage = w;

}

public void setFullTime(boolean f){

this.fullTime = f;

}

public void printDetails(){

String out="";

if(this.fullTime){

out=out+"The name of the worker is "+this.name + ". The worker id is "+this.workerId;

out=out+". The wage of the worker is $"+this.wage+" per hour. The employee is not full-time.";

}

else{

out=out+"The name of the worker is "+this.name + ". The worker id is "+this.workerId;

out=out+". The wage of the worker is $"+this.wage+" per hour. The employee is full-time.";

}

System.out.println(out);

}

}

public class temp{

public static void main(String[] args)

{

Worker w1 = new Worker("John Smith", 123456, 25.40, false);

w1.printDetails();

Worker w2 = new Worker("Aidra Rhodes", 654321, 2500, true);

w2.printDetails();

}

}

---------------------------------------------------------------------------------

Assignment Q3

To be completed in week 3

Create a new BlueJ project called YourFirstNameLastName-A1Q3.

Create a class, Assignment, that contains the following four fields:

• A String called StudentName

• A double called assignmentMark (which will store the mark each assignment is worth

e.g. for this assignment that you are doing right now the value would be 20)

4

• A double called studentMark (stores the mark the student gets in the assignment e.g.

15)

• A String called grade

Make sure you write a description of your new Class in the comments, with your name as

author and give the version as the date you last worked on this exercise.

Define a constructor that takes and sets the studentName, studentMark and

assignmentMark.

Also define a constructor that takes no parameters and sets the assignmentMark to 100.

Create an accessor and mutator for studentMark. The mutator should not let the

studentMark be set a value greater than the assignmentMark (as the student cannot get a

mark higher than the assignment is worth) or less than 0. If the user tries to set a value that

is not valid a suitable error message should be displayed.

Create a method that calculates the grade for the student. You will need to work out how

many percent the student scored in the assignment.

If the student scored:

• Less than 50% the grade will be fail

• 50% - 64% the grade will be pass

• 65% – 74% the grade will be credit

• 75% – 84% the grade will be distinction

• Greater than 85% the grade will be high distinction

For example, if assignmentMark is 30 and studentMark is 15, the percentage will be 50% so

the grade will be set to pass.

Define an accessor method to return the value of grade.

Ans.=>

public class Assignment{

String studentName;

double assignmentMark;

double studentMark;

private String grade;

public String getGrade() {

return grade;

}

public double getStudentMark() {

return studentMark;

}

public Assignment(){

this.studentName = "abc";

this.studentMark = 30;

this.assignmentMark = 100;

}

public Assignment(String studentName, double assignmentMark, double studentMark){

this.studentName = studentName;

this.studentMark = studentMark;

this.assignmentMark = assignmentMark;

}

public void setStudentMark(double studentMark){

if(studentMark < 0 || studentMark > this.assignmentMark)

System.out.println("Student mark should be between 0 and " + this.assignmentMark);

else

this.studentMark = studentMark;

}

public void calculateGrade(){

double percentage = this.studentMark / this.assignmentMark \* 100;

if(percentage < 50.00)

this.grade = "Fail";

else if(percentage >= 50.00 && percentage<= 64.00)

this.grade = "Pass";

else if(percentage >= 65.00 && percentage<= 74.00)

this.grade = "Credit";

else if(percentage >= 75.00 && percentage<= 84.00)

this.grade = "Distinction";

else if(percentage >= 85)

this.grade = "High Distinction";

}

}

----------------------------------------------------------------------------------------------------------------------------------

Assignment Q4

To be completed in week 4

Create a new BlueJ project called YourFirstNameLastName-A1Q4.

Create a class called ListOfNames, that has one ArrayList field called names, which holds a

collection of Strings (each string is a male or female name in Upper case e.g. PETER).

Make sure you write a description of your new Class in the comments, with your name as

author and give the version as the date you last worked on this exercise.

5

Define a constructor that initialises the ArrayList. Note that you can also add any other

initialisations that you feel are relevant.

Create methods to add elements, remove elements and get the number of elements in the

collection. Add a test for all three of these methods to check whether the operation was

successful and print a message letting the user know if it was or was not.

Create a method called printNames. This method should loop through the collection and print

out the elements (each String on a new line) as determined by the following rules:

• If the string contains any vowels (A, E, I, O and U), the method should print "The name

" + the value of the String + " contains vowels, and the vowels are:" + list of the vowels

in the string. The string may have more than one vowels. For example, if the name is

EMMA it would print: The name EMMA contains vowels, and the vowels are: E, A

• If the string contains duplicate characters, the method should print "The name " + the

value of the String + " has the following duplicate character(s):" + list of the duplicate

characters in the string. The string may have one or more duplicate character. For

example, if the name is ANNABELLA, it would print: The name ANNABELLA has the

following duplicate character(s): A, N, L

• If the string contains any vowels (A, E, I, O and U), and it has duplicate characters, the

method should print "The name " + the value of the String + " contains vowels and has

duplicate characters". For example, if the name is LARISSA it would print: The name

LARISSA contains vowels and has duplicate characters.

• If none of the above criteria is met, then the method should print the String element

in lower case. For example, if the name is SKY, it would print: sky

Once you have finished your project, open the terminal window in BlueJ and turn on record

method calls. Create a new ListOfNames object, and then add at least ten (10) Strings using

the add method you wrote. You must have:

• A String that contains vowels

• A String that has duplicate characters

• A String that contains vowels and has duplicate characters

Demonstrate removing an element using the remove method you wrote, and then find the

number of elements using the method you wrote that gets the number of elements. Finally,

run your printNames method.

Copy all your calls into a text file called YourFirstNameLastName-A1Q4-example.txt and save

it in your BlueJ project folder.

Ans.=>

import java.util.ArrayList;

public class ListOfNames {

private ArrayList<String> names;

public ListOfNames() {

names = new ArrayList<String>();

}

public void add(String name) {

name = name.toUpperCase();

names.add(name);

System.out.println("Added name " + name);

}

public void remove(String name) {

name = name.toUpperCase();

if(names.remove(name))

System.out.println("Removed name " + name);

else

System.out.println("Not found name " + name);

}

public int size() {

return names.size();

}

public void printNames() {

for(int i = 0; i < names.size(); i++) {

String name = names.get(i);

String vowels = "";

String duplicates = "";

for(int j = 0; j < name.length(); j++) {

char c = name.charAt(j);

if(c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U') {

if(vowels.indexOf(c) == -1)

vowels += ", " + c;

}

for(int k = 0; k < j; k++) {

if(duplicates.indexOf(c) == -1 && name.charAt(k) == c)

duplicates += ", " + c;

}

}

if(vowels.equals("" ) && duplicates.equals(""))

System.out.println(name.toLowerCase());

else if(vowels.equals(""))

System.out.println("The name " + name + " has the following duplicate character(s): " + duplicates.substring(2));

else if(duplicates.equals(""))

System.out.println("The name " + name + " contains vowels, and the vowesl are: " + vowels.substring(2));

else

System.out.println("The name " + name + " contains vowels and has duplicate characters");

}

}

public static void main(String[] args) {

ListOfNames list = new ListOfNames();

list.add("EMMA");

list.add("LARISSA");

list.add("PETER");

list.add("SKY");

list.add("ANNABELLA");

list.add("MICHAEL");

System.out.println("List has " + list.size() + " names");

list.printNames();

}

}

---------------------------------------------------------------------------------------------------------------------------------------

Assignment Q5

To be completed in week 5

Part A:

Imagine you need to write a program for a 24-hour clock with hours, minutes and seconds.

Write a Java program in BlueJ with a method that prints all possible times the clock could

display starting at 00:00:00 through to 23:59:59 when all the three numbers are the same

(e.g. 01:01:01, 02:02:02, 13:13:13 and so on)

Part B:

Write a second method that takes three (3) parameters – hours, minutes and seconds. This

method will print out all of the possible times the clock could display from one hour before

the time passed to the method till one hour after the time passed to the method when all the

three numbers are even (e.g. 12:20:00 or 12:20:02, NOT 12:20:01). For example:

If the method was passed the following values:

Hour = 11

Minutes = 23

Seconds = 44

The method would print all the times the clock could display from 10:23:44 until 12:23:44

when all the three numbers (hours, minutes and seconds) are even. The first printed time

would be 10:24:00, and the last one would be 12:22:58)

Ans.=>

import java.util.Scanner;

public class Time2 {

public static void main(String[] args) {

String first ="10:24:00";

String last = "12:22:58";

int hours ,minutes,seconds;

Scanner sc = new Scanner(System.in);

hours = sc.nextInt();

minutes = sc.nextInt();

seconds = sc.nextInt();

if(hours %2==0 && minutes%2==0 && seconds%2==0){

System.out.println("first printed time time:"+first+" last printed time:"+last);

}

else {

System.out.println("hours or minutes or second are not even");

}

}

}

--------------------------------------------------------------------------------------------------------------------------------------