**OOP Problem Set 1 : printf, String.format(), JTextAreas, Fonts, Color and revision of loops, user-defined methods and arrays.**

Answer as many of the problems as possible in the lab hours allocated and be sure to attempt the rest in your own time. Use input dialogs for all user input. Include user defined methods in any many places as you think appropriate.

1. **While loop**, and using printf() / String.format() to round to 2 decimal points.

Look up the internet to see what the exchange rate for GB£ is today (in euro per GB£), then write a program which converts from GB£ to euro. The program should ask you to enter the exchange rate once, then the program should contain a while loop which repeats automatically for an arbitrary number of conversions, terminating when a sum of £0 is entered. Both GB£ and euro should be displayed correct to 2 decimal places with each loop iteration, and each output dialog should contain the exchange rate in its title. How would you describe the type of while loop needed in this program?

2. **For loop** and JTextArea.

Write a program to produce the following table using a **for loop**. Send the output to a JTextArea and display the JTextArea within a message dialog. Don’t worry about aligning the layout. See DialogDemoA for an example of using a JTextArea. There are 12 inches in one foot and 3 feet in one yard.

Yards Inches

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

1 36

2 72

. .

. .

10 360

3. **Input dialogs**, printf() and String.format().

Write a program to calculate the cost of carpeting a room wall to wall. The user should be asked for his/her name, the length of the room, breadth of the room and cost per square metre of the carpet, each to be entered via an input dialog. The program should then generate a quotation for the customer containing information in the following format. Suggestion: send the output initially to System.out: when you have that working, transfer it to a JTextArea within a message dialog. You will need to set the Font of the JTextArea to a monospaced font.

Quotation for Joe Soap

Length of room: 3.50 m.

Breadth of room: 2.55 m.

Total area of the room: 8.93 sq. m.

Cost per square metre of carpet: 33.50 euro

Total cost of carpet: 298.99 euro

The last 5 lines of output should be aligned properly as shown. The easiest way to ensure the alignment is something as follows:

System.out.printf(" %-20s%.2f m.\n %-20s%.2f m.\n","length",len,"breadth", br);

where the fixed text is allowed for in the formatting string as a String of specific width.

4. Discovering some **methods from the String class**; identifying variables, objects and methods used. Write a program that asks the user to enter their full name, including first name, middle name if any, and surname, as one String. The program should then output the following information:

* The number of characters in the name *(use the length() method)*
* The initial of the firstname *(declare a char variable to hold the initial, and use charAt(0))*
* The name as entered, but entirely in capital letters *(use toUpperCase())*
* The surname on its own *(use lastIndexOf() to find the last space, then use substring)*

5. **Methods from class Math** and arithmetic in java.

You are given a quadratic equation as follows:



and you need to find its roots i.e. the value(s) of x which make the left hand side of the equation equal to zero.

Using the formula: 

where a is the coefficient of x2 in the equation (1 in this case), b is the coefficient of x (6) and c is the constant part (-16), find the roots of the above quadratic equation. **Hint**: use the sqrt() method in Java’s Math class and declare and calculate two root variables to account for the + and - cases in the formula. Be careful about arithmetic precedence.

6. **User-written methods**.

Write your own user-defined method called *cube()* which is passed an integer value and returns the cube of it (i.e. it will return an integer also). In your main() method you should have this method called repeatedly (using a loop) to display a table of values between 1 and 15 inclusive along with their cubes as indicated in the screen capture below.



Draw a class diagram for your solution.   
  
Pushing the boundaries: for a second version of your solution, take your cube() method out of the class which contains your main and store it in a new class called MyMethods.java. Then invoke it within your main by prefixing it with MyMethods, as in MyMethods.cube(x) instead of cube(x). Write another application which asks you to enter the length of the side of a cubic container for face cream, which will invoke MyMethods.cube() to calculate the volume of the cube. If the resultant volume is greater than 100 cm3, your program should output “You can’t take this on board, put it into your check-in luggage”. Draw a VOPC diagram showing the two classes now involved in your system, and the methods contained in each.

7. **User-defined methods**, booleans and the „%‟ operator. Write a program that asks the user to input an arbitrary number of integer values (terminated by –1) and passes them one at a time to a user-defined method called *isEven()* which uses the remainder (%) operator to determine if an integer is even. The method should take an integer argument and return the boolean value “true” if the integer passed in is even and “false” otherwise. In the main() method the actual outcome is displayed e.g. “5 is odd” or “10 is even” etc.

8. **Arrays.** Write a program that asks the user to input a list of 10 integers. Store these values in an array of the required size and output the 1st and 5th values in the array. Use a for loop to read in the values efficiently. Your output should appear as indicated below.   
  


9. **Array Algorithms** total, average, biggest, selective count. You have taken a job working at the cattle mart for the summer. Write a program that declares an array of 10 doubles and initializes it to the weight in kilos of 10 animals. Your program should output the following:

(a) All the weights, along a line

(b) the number of animals weighing less than 250 kilos

(c) the percentage of animals weighing at least 400 kilos

(d) the lightest animal

(e) the average weight of all the animals

A sample output from the program might be as follows, for the case of 5 values. Note the average is displayed correct to 1 decimal place.   
  
Weights: 234.2, 123.0, 344.8, 455.4, 150.6

Number under 250 kg: 3

Percentage over 400 kg: 20%

Lightest animal: 123.0 kg

Average weight: 261.6 kg