For the given tasks (see Appendix), you are required to produce a logbook document using the Java programming language.

**Please do not submit zipped files.**

**Marking grid**:

|  |  |
| --- | --- |
| **Title** | **Logbook coursework** |
| **Task details summary** | From the tasks (given in the appendix section), you will be required to implement, test and document solutions to the given tasks using Java programming language.  You will submit your logbook in a word or pdf format with clear evidences in terms of screenshots of the code and output.  Program documentation should include:   * A header page with appropriate identification * A contents list * Introduction of your application and the features that you have implemented * Program listing with clear annotations to illustrate features implemented * Screen shots to illustrate testing/program output for each of the required functionality |

**Appendix**

The tasks listed below provide an informal specification of programs that you are asked to develop, test and document. All these tasks must be included in the logbook with appropriate coding standards and documentation.

**Tasks:**

**Task 1:** Write a Java program to declare integer, long and float data types. Following the methods of type casting, demonstrate implicit and explicit type casting amongst the declared data types.

**Task 2:** Write a Java program using string manipulations.

String p = Enter user input

String q = Enter user input

String r = Enter user input

Write code to work out and display:

* Concatenation (p+q+r) in capital letters
* Concatenation of (r + p) in lowercase
* total number of characters within (p+q+r)

**Task 3:** Create an interest calculator that will prompt the user to input a starting balance (£1000) and an interest rate (5%) on the starting balance. Work out the balances after the first, second, third year and fourth year.

**Task 4:** Write Java program to allow the user to input his/her age. Then the program will show if the person is an infant, child, a teenager or an adult.

**Task 5:** Write a Java program that prompts the user to enter the number they wish to see the multiplication table of. In addition, the program should prompt the user to enter the length of multiplication table (i.e. counter) and displays the table on the screen.

**Task 6:** Write a Java program to make a calculator. The calculator should add, subtract, multiply and divide the numbers inserted by the user. Feel free to add more operations to your calculator, such as power, sqrt etc. (Hint: Use Switch)

**Task 7:** Create a class called GuessNum. Write code that randomly generates an integer between 1 and 100, inclusive. The program prompts the user to enter a number continuously until the number matches the randomly generated number. For each user input, the program tells the user whether the input is too low or too high, so the user can choose the next input intelligently. Display the number of attempts made for the user to guess the number.

**Task 8:** Write a program that sets a password as “changeme” and asks the user to enter the password. The user gets only 5 attempts to get the password right otherwise the program should display “Access denied. Please contact IT services to reset your password”. Once correct password is entered within 5 attempts the program should say “Password Accepted”.

**Task 9:** Write a Java method to check whether a year (integer) entered by the user is a leap year or not.

Expected Output:

Input a year: 2017

False

**Task 10:** Write a Java program to calculate the average value of array elements. The array contains 10 numeric values of your choice.

**Task 11:** Create a new NetBeans project. In this project, in a separate file from the main() method, create a second Java class called Product, and in a third file create a third class called Book.

Your Product class should have five private data fields:

A variable productID, of type String, to store the product ID

A variable price, of type int, to store the price of the product

A variable primeEligible, of type boolean, to indicate whether or not the product is eligible for Amazon Prime

A variable numberInStock, of type int, to store the number of this product currently in stock

A variable dateAdded, of type Date, to store the date when the product was added to the Amazon website

1. Write a parameterised constructor for the Product class, to initialise its data fields, and write setter and getter methods for each of the data fields.

1. Add the following private data fields to the Book class:

A variable title , of type String, to store the book title

A variable author, of type String, to store the name of the book’s author

A variable numPages, of type int, to store the number of pages in the book

A variable publisher, of type String, to store the name of the publisher

A variable publicationDate, of type String, to store the date when the book was published.

1. Write a parameterised constructor for the Book class, to initialise its data fields - the constructor should call the Product class constructor by using the super keyword. Then write setter and getter methods for each of the data fields.

1. In the main() method, create three objects of type Book – look on the Amazon website to get the information you need to pass to the constructor. Then use the getter methods to display the information about the products, including the data fields that were inherited from the Product class. Use the setter methods to change some of the data in your objects.

**Example Output:**

