VAAL UNIVERSITY OF TECHNOLOGY
Inspiring thought. Shaping talent.

ASSIGNMENT 1

FACULTY OF APPLIED & COMPUTER SCIENCES

DEPARTMENT OF ICT

DIPLOMA: INFORMATION AND COMMUNICATION TECHNOLOGY

SUBJECT	:	DEVELOPMENT SOFTWARE 3.1
SUBJECT CODE	:	ASDSX3A
FORMATIVE ASSESSMENT	:	
DUE DATE	:	15 MARCH 2025
DURATION OF TEST	:	
EXAMINER(S)	:	MRS S MOYO
MODERATOR(S)	:	MR N SOGANILE

REQUIREMENTS : NetBeans version 8.2 and above

INSTRUCTIONS:

1. Answer ALL the questions
2. Submit your zipped NetBeans project via an assignment link on VUTela**.**

MARKS: Total = 100
Full marks = 100

Problem Description

You are working for the online store takealot.com as a software developer. You are on the team that is busy developing software for the books section of the online store. The store is selling both printed (hard copy) books as well as ebooks that can be downloaded. The **information that is needed for ebooks** is the **author**, **title**, **ISBN** number as well as **file size**. File size is given in kilobytes and is an integer value.

The **information needed for printed books** is **author**, **title**, **ISBN** number, **number of pages** as well as **weigh**t. Weight is given in kilograms and is a real value.

ISBN numbers are unique identication numbers allocated to books. An ISBN number is a string of 10 digits. The last digit may also be the character `B'.

The first digit of the ISBN number indicates the type of the book: **Printed books ISNB number starts with the digit 1** and **electronic books ISBN starts with the digit 0**.

Sample data is shown below.

Example data:

Printed books				
Title	Author	ISBN	Pages	Weight(kg)
Programming Logic & Design	Joyce Farrel	1432456344	950	1.1
Computer Programming	Alexander Bell	198734561B	1850	1.95
Core Java	Cay S. Horstmann	1367823245	1450	1.5

eBooks			
Title	Author	ISBN	Size(kb)
Programming Logic & Design	Joyce Farrel	0432456345	33178
Murach's Java Programming (6th Edition)	Joel Murach	067001617B	8734
Life Orientation	S. Mahuluhulu	0870636141	577

QUESTION 1: Inheritance and polymorphism

From the given information, identify all necessary classes with their attributes, constructors and methods. Two concrete classes **EBook** and **PrintBook** and one abstract class, **Book**, should be amongst the identified classes. You should also create at least one interface, of your own, that should be implemented by your abstract class.

Both the EBook and PrintBook classes must contain the method **getSizeDetails.** The **getSizeDetails** method in the EBook class must return the title and file size of the book as a string as in **Figure 2** while in the PrintBook class it must return the title, number of pages and weight of the book as a string as in **Figure 1**.

Produce code for all the identified classes.

Abstract methods must be used to ensure the re-use of as much code as possible and to ensure polymorphic behaviour. Therefore no code or instance variables may be duplicated.

Instance variables must be declared as below. DO NOT declare any extra instance variables.

```
private String title
private String author
private String isbnNo
private int noOfPages
private float weight
private int fileSize
```

Coding the setIsbnNo method:

ISBN numbers must have exactly 10 DIGITS. The last digit may also be the character `B' or ‘b’. Validity testing of the ISBN numbers must include testing whether the length of the given ISBN number is correct and testing whether the ISBN number contains only digits with an optional character 'B' in the place of the last digit.

Note that an invalid ISBN number may have an incorrect length AND contain non-digits. You therefore need to test for this situation as well.

The set method (in one or more of the book classes) for the ISBN number instance variable must test the validity of the ISBN number that it receives. If the ISBN number is invalid the set method must throw an

IllegalAgumentException. Different messages must be given for each of the three situations described above - see the output given below.
Code the **toString** methods of the EBook and PrintBook classes to display attributes as in **Figure 4** and **Figure 5** respectively. These toString methods must access the methods of the super class using the super keyword. DO NOT code a toString method for the super class.

Exception test output:

```
Invalid ISBN:  Must be 10 characters.ISBN = 192156844

: Invalid ISBN: The first nine characters must be digits only. ISBN = 032156B840

Invalid ISBN: The last character may only be B or b. ISBN = 198734561K
```

```
Printbook: Programming Logic & Design, 950, 1100g
Printbook: Core Java, 1450, 1500g
```

Figure 1

```
EBook: Life Orientation, 577 KB
EBook: Murach's Java Programming( 6th Edition), 8734 KB
```

Figure 2

QUESTION 2:

Using the JUnit testing testing framework:

1. verify if the **setIsbnNo** mutator is validatng the input correctly or not by creating the following test methods in the **ISBNTestClass** class and using the given account numbers. The class is found in the **vut.test** package in the project's **Test Packages**. If the class was not provided, create it. Create the following test methods and use the ISBN number appended to the name of the method to test the setIsbnNo method.

    a. testSetIsbno067001617B()
    b. testSetIsbno1367823245()
    c. testSetIsbno198734561B()
    d. testSetIsbno192156844()
    e. testSetIsbno032156B840()

f. testSetIsbno032156840b()
g. testSetIsbno198734561K()

QUESTION 3: GUI Appliction

The application consists of only one screen shown in **Figure 3.** Code the BookEntryScreen form class as follows:

1. Create a single array list named **bookList** that must be able to contain both EBook and PrintBook objects.
2. Create a private method called **fillCmboPages()** to populate the **cmboPages** control with integers from 400 to 2000 in increments of 50.
3. Create a private method called **fillCmboWeight** to populate the **cmboWeight** control with integers from 1000 to 2000 in increments of 5.
4. Code the Save Book button as follows:
    a. Read the ISBN number from the text field into the isbnNo variable,
    b. Use a try.. catch block to handle exceptions that may be thrown by the classes.
    c. Use an if structure to check the nature of the value in the isbnNo variable. If the value in the isbnNo variable cannot be associated with neither EBook nor PrintBook objects, display an error message to the user using a message box otherwise create the appropriate object using either the EBook or PrintBook class and store the object in the array list bookList. The creation of the object and adding the object to the array list must be done in a single code statement.
    d. Display the recently created object from the array list onto the text area.
5. Code the **View Weight of Books** button to display the list of the titles and weights of all the printed books in the array as in **Figure 6.** Use the getSizeDetails method.
6. Code the **View Size of Books** button to display a list of the titles and size information of all the books in the array as in **Figure 7**. Use the getSizeDetails method.

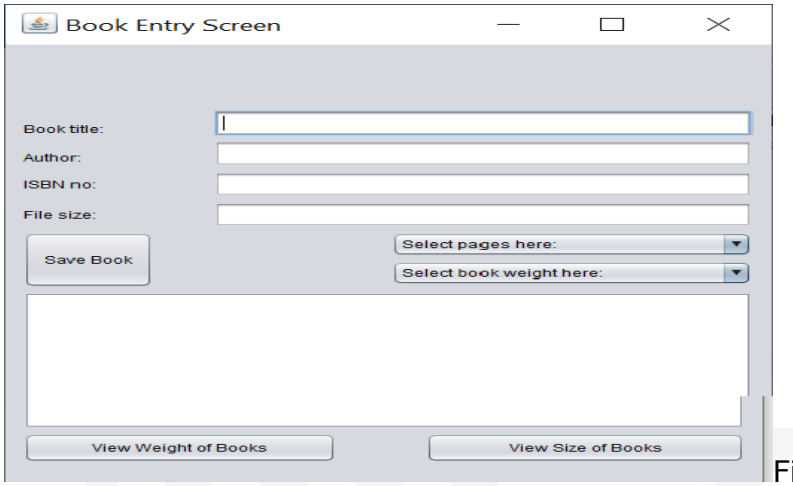

Figure 3

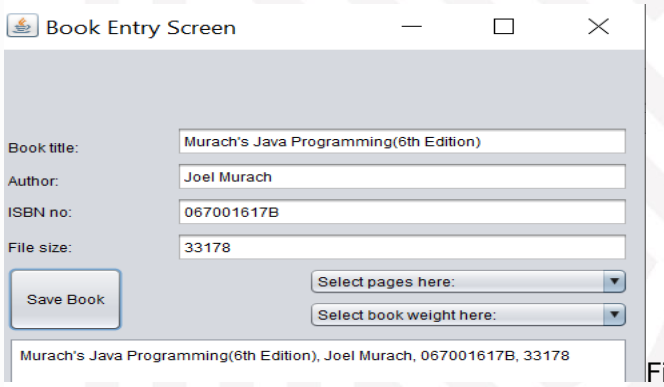

Figure 4

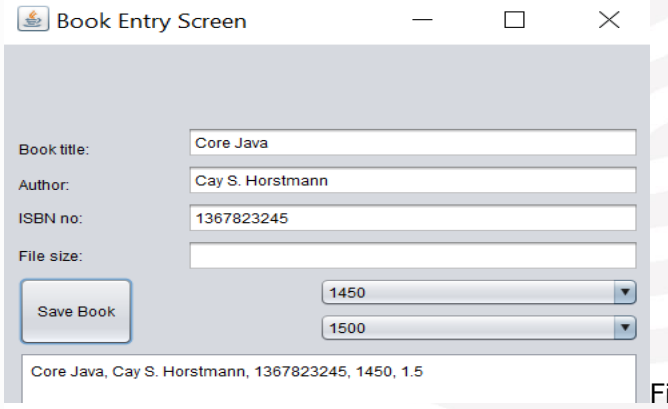

Figure 5

Figure 6

Figure 7

*******The End ****Dala what you must*********