

**PROGRAMME :**

**DIPLOMA IN INFORMATION TECHNOLOGY**

**(DIGITAL TECHNOLOGY)(DDT)**

**COURSE:**

**DFP30243- OBJECT ORIENTED PROGRAMMING**

|  |  |  |
| --- | --- | --- |
| **ASSESMENT** | **LABORATORY TASK 3** | |
| **NO** | **REGISTRATION NO** | **NAME** |
| 1. |  |  |
| 2. |  |  |
| **PROGRAMME** |  | |

INSTRUCTIONS :

1. Answer **ALL** the questions.
2. Submission Date : ……………………………………………………

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| **CODE / COURSE** | **DFP30243–** OBJECT **ORIENTED PROGRAMMING** | **LABORATORY TASK** | **~~1 / 2~~ / 3 ~~/ 4~~** | |
| **PROGRAM / CLASS** |  | **DURATION** | **120 MINUTES** | |
| **STUDENT’S NAME** |  | **CLO** | **1P** |  |
| **REG. NO.** |  | **TOTAL MARKS** | **/50** | |
| **LECTURER’S NAME** | **PN. HAZLEENA BINTI OSMAN** |

**CLO1:** Construct Object Oriented Programming concept and exception handling in Java programming (P4, PLO3)

**Learning outcomes:**

Upon completion of this lab, students should be able to:

1. Construct Inheritance
2. Construct Polymorphism

**INSTRUCTION: ANSWER ALL QUESTIONS.**

**QUESTION 1**

Write a java inheritance program using *abstract class* based on the Class diagram given. The sample output is as below: -

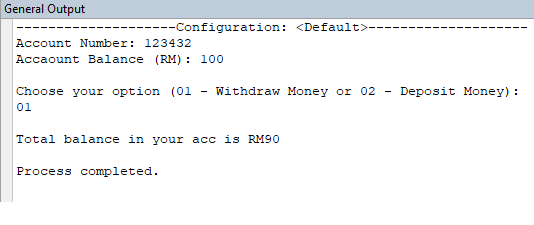
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | *AccBank*  *(Abstract Class)* | | +accNumber : int  +accBalance : int  +optionAction : int | | +accDisplay():abstract void  +accInfo():void |  |  | | --- | | *02 - deposit*  *(Subclass Class)* | | *+depositMoney : int* | | +accDisplay ():void |  |  | | --- | | *01 - withdraw*  *(Subclass Class)* | | *+withdrawMoney : int* | | +accDisplay ():void | |

**Figure 1**

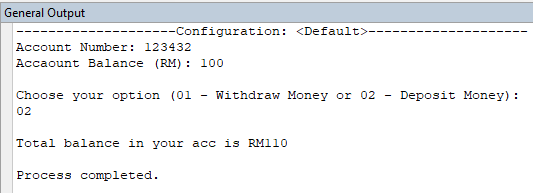
Figure 1 illustrates the process flow showing the interconnections relationship between the various entities. It is your responsibility as a developer to create a Java programme that follows the structure shown in figure 1. These are the requirements for to follow:

1. accDisplay() must be declared as an abstract method
2. The variable accBalance must be initialised with a value of 100.
3. Set the variable accNumber to any number.
4. Use a bufferedReader to receive user input.
5. In order to withdraw, the customer should select option 01, while making a deposit, should select option 02.
6. Process of money withdrawal and deposit
   1. Only RM10 can be withdrew, and the subtraction method of calculation is used.
   2. Only can deposit RM100 to the account, and calculations may persist with additional mathematical
7. Last, using accDisplay(), display the current accBalance amount in the class withdraw and deposit procedure.

**Sample Output:**



For Withdraw process



For Deposit process

|  |
| --- |
| 1. **Source code**   **\*Snip and paste your source code here. (Make sure it is snipped from your text editor/ java platform). Refer Appendix 1 for an example.** |
| 1. **Output**  * **Snip and paste your output here. (Make sure it is from your command prompt/ java platform). Refer Appendix 1 for an example.** |