Odd Semester (2020)



**BINUS UNIVERSITY**

**BINUS INTERNATIONAL**

**Assignment Cover Letter**

**(Individual Work****)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | |  | |  | |
| **Student Information**: **Surname** | | | | | **Given Names**  **Veronika** | | **Student ID Number**  **2001586161** | |
| 1. | | **Stephanie** |  | |
|  |  |
| **Course Code** | **: COMP6335** |  |  | | **Course Name** | | **: Introduction to Programming** | |
| **Class** | **: L1AC** |  |  | | **Name of Lecturer(s)** | | **:** 1. Bagus Kerthyayana | |
|  |  |  |  | |  | | 2. Stavin | |
| **Major** | **: CS** |  |  | |  | |  | |
| **Title of Assignment**  (if any) | : Interest and Depreciation | |  |  | |  | |  | |
| **Type of Assignment**    **Submission Pattern** | **: Final Project** |  |  | |  | |  | |
| **Due Date** | **: 6-11-2016** |  |  | | **Submission Date** | | **: 6-11-2016** | |

The assignment should meet the below requirements.

1. Assignment (hard copy) is required to be submitted on clean paper, and (soft copy) as per lecturer’s instructions.
2. Soft copy assignment also requires the signed (hardcopy) submission of this form, which automatically validates the softcopy submission.
3. The above information is complete and legible.
4. Compiled pages are firmly stapled.
5. Assignment has been copied (soft copy and hard copy) for each student ahead of the submission.

# Plagiarism/Cheating

BiNus International seriously regards all forms of plagiarism, cheating and collusion as academic offenses which may result in severe penalties, including loss/drop of marks, course/class discontinuity and other possible penalties executed by the university. Please refer to the related course syllabus for further information.

# Declaration of Originality

By signing this assignment, I understand, accept and consent to BiNus International terms and policy on plagiarism. Herewith I declare that the work contained in this assignment is my own work and has not been submitted for the use of assessment in another course or class, except where this has been notified and accepted in advance.

Signature of Student: (Name of Student)

1. Veronika Stephanie

**“Interest and Depreciation”**

**Name :Veronika Stephanie**

**ID :2001586161**

1. **Description**

**The function of this program:**

This program is meant to help people to calculate compound interest, single interest, book value depreciation, and good value depreciation. Especially for the depreciation, the program will tell you the exact month when your things do not have any value anymore.

**II.a. Design/Plan**

**Project’s Hierarchy Chart**

IntAndDep

class

Single Interest

Compound Interest

Depreciation Book Value

Depreciation

Good Value

Interest

Depreciation

Interest

menu

Depreciation

menu

Main

Menu

**II.b. Explanation of Each function**

**Main Menu: ( *main.cpp* )**

1. **Outside int main ():**

* Holds other function declaration :*void interestChoices ()*; *void depreciationChoices ()*; *double compoundInterest (double, double, double, double)*;
* Holds *void exitchoice()*: this function is meant to be used in other cpp and header file if it is necessary, it contain statement and option ( yes or no ) whether the user wants to continue the program or exit the program.

1. **Inside int main ():**

* Using a do while loop to show the option whether the user want to go to the depreciation section, interest section, or to exit the program.
* If the user input choose the interest section, the program will run the *interestChoice()* function.
* If the user input choose the interest section, the program will run the *depreciationChoice()* function.
* If the user input chooses to exit, the program would be terminated.
* The input validation (choice input could not more than 3 or less than 1) using while loop.

**Interest Menu: ( *interest menu.h* )**

* Declaration of *void exitchoice()* because the exit option is needed, so it needs to be declared beforehand.
* *void interestChoices()* ask the user to input the choice whether the user wants to go to the compound interest, the single interest, or go back to the main menu.
* After the input, the program will continue to check the condition. To check the user’s input, this project used the while loop to check whether the user input number less than one or more than three.
* After the user input the option, if the input is compound interest option, user is asked to input the capital, month, and the percentage, then the input will be transferred to the compound interest constructor and the constructor will run. Otherwise, if the input is single interest option, user is asked to input the capital, month, and the percentage, then the input will be transferred to the single interest constructor and the constructor will run. Else if the user input the input is go back to the main menu, the *return* statement will help us to return to the main menu in *main.cpp*.
* After the constructor run ( calculated the interest according to the formula in each of the constructor and showing the output ), the *exitchoice()* functions are placed after the constructor, so it will asked whether the user wish to exit or to continue the program. If the user wishes to continue, it will return to the interest and depreciation menu that is looped (located in: *main.cpp* ).

**Interest: (***interest.cpp and interest.h*)

***Interest.h:***

* Contains of a class named interest that inherit IntAndDep class.
* Declaration of class member functions, constructor, and attribute (class specification):

1. Private:

* compoundInt;

1. Public:

* Interest(double p, double c, double m);
* double getSingleInterest ()const;
* Interest (double, double, double, int);

***Interest.cpp:***

* Contains the function definitions and the constructor definition (class implementation):

1. Interest(double p, double c, double m), constructor to set the percent, capital, and month from user’s input.
2. double getSingleInterest ()const, to calculate the single interest form the data that have been transferred to the private member that inherited from IntAndDep class.
3. Interest (double, double, double, int), constructure to calculate and get the compound interest form the data that have been transferred to the private members that inherited from the IntAndDep class and using a pointer to dynamically allocated the memory for the value of each month.

**Depreciation Menu: (*depreciation menu. h*)**

* Declaration of *void exitchoice()* because the exit option is needed, so it needs to be declared beforehand.
* *void deprecitationChoices()* ask the user to input the choice whether the user wants to go to the compound interest, the single interest, or go back to the main menu.
* After the input, the program will continue to check the condition. To check the user’s input, this project used the while loop to check whether the user input number less than one or more than three.
* After being validate, if the input is deprecitation good value option, user is asked to input the capital, month, and the percentage, then the input will be transferred to the depreciation good value constructor and the constructor will run. Otherwise, if the input is depreciation book value option, user is asked to input the capital, month, and the percentage, then the input will be transferred to the depreciation book value constructor and the constructor will run. Else if the user input the input is go back to the main menu, the *return* statement will help us to return to the main menu in *main.cpp*.
* There are try and catch statement below each of the depreciation constructor. The try block purpose is to check the if condition in each of the get depreciation function (*getDepreciationGoodValue()* to check the good depreciation and *getDepreciationBoodValue()* to check the book depreciation) then if the condition makes exception is thrown, the function would be terminated and the catch block would handle the exception and run the statement inside the catch block if any.
* After the constructor run ( calculated the depreciation according to the formula in each of the constructor and showing the output ), the *exitchoice()* functions are placed after the constructor, so it will asked whether the user wish to exit or to continue the program. If the user wishes to continue, it will return to the interest and depreciation menu that is looped (located in: *main.cpp* ).

***Depreciation.h:***

* Contains of a class named Depreciation that inherit IntAndDep class.
* Declaration of class member functions, constructor, and attribute (class specification):

1. Public:

* Depreciation(double p, double c, double m);
* double getDepreciationGoodValue ()const;
* double getDepreciationBookValue ()const;

***Depreciation.cpp:***

* Contains the function definitions and the constructor definition (class implementation):

1. Depreciation(double p, double c, double m), constructor to set the private member value using the user’s input.
2. double getDepreciationGoodValue ()const, calculate the depreciation good value using the private members (inherited from IntAndDep class) that have been set by the constructor, outputting the value for each month and throwing an exception if the value reach zero or below zero.
3. double getDepreciationBookValue ()const, calculate the depreciation good value using the private members (inherited from IntAndDep class) that have been set by the constructor, outputting the value for each month and throwing an exception if the value reach zero or below zero.

**IntAndDep class: (*IntAndDep.h*)**

* Contains a class named IntAndDep, that consist of :

1. Protected member:

* percent;
* capital;
* month;

1. Public member:

* double getPercent()const, return the value of percent.
* double getCapital()const, return the value of capital.
* double getMonth()const, return the value of month.

**III.a. Lessons that Have Been Learned**

6th of October,

The first time I uploaded my final project progress using Git Bash. The source code is still simple, only using cout for output and cin for user’s input. Although it is very simple it is still an important lesson, it is a practice to use the right symbol for cin and cout. Also, to learn the basic things that are needed for a program in order to run, for example include library, using namespace, and int main.

10th of October,