CS52(SMC Spring 2019) – Lab2 (See due date at Canvas)

Rules for Developing Software Using Functions

Below is NOT a complete list of rules. Additional rules may be stated in lecture or are inferred by output shown in test runs. In any case it is program writer's duty to find out all program specifications, which are NEVER presented to them as neat, and nice bundle. Real life software development is messy. We do our best to design a clean path for you. But Computer Science department wishes that you must become aware, that creativity in software is to design and produce neat, clean, and efficient software from messy specifications. Here are the rules.

1. Programmer is NOT allowed to change function signatures. That includes following

A. Function name. This includes even case of letters in the function name. For example, if function name is getNetSalary, then name getnetsalary breaks the specification and is not accepted.

B. Data types and order or argument list, as well as parameter passing mechanism for each parameter. The latter includes pass by value or reference. If passed by reference then whether passed by const reference or not?

For example if function header is:

```
void print (const string & firstName, const string & lastName, int age);
```

Then below are some examples of violation of specification.

```
void print (string & firstName, string & lastName, int age);
void print (const string & firstName, string & lastName, int age);
void print (const string & firstName, const string & lastName, long age);
void print (int age, const string & firstName, const string & lastName);
void print (const string & firstName, int age, const string & lastName);
void print (const string & firstName, const string & lastName, const int  & age);
void print (const string & firstName, const string & lastName, int & age);
void print (const string & firstName, const string & lastName, int age, double
salary);
```

2. Although return type is not the part of signatures, programmer is NOT allowed to change that either.

3. If it is specified that function cannot output to console or file or cannot take any user input from file or keyboard, then that specification Must be obeyed.

As mentioned earlier, this is ONLY a partial list. We are in the process of preparing a coding manual that will have all details. Till then please rely upon specification written in assignments, communicated to you in lectures, e-books, textbooks, and output from our programs shown in your assignments.

Bigger Picture/Prologue

Since invention of first computer called ENIAC (1949), software has come a long way. Table below gives typical number of lines of codes in modern software in different institutions and products. The reason we give lines of codes for a software, because just like the number of parts in a machine signify its complexity, lines of code gives the similar metrics for a software.

Program/Software	**Approximate Lines of Code**
Typical Hello World Program	**1 - 10**
Implementation of stacks and queue data structure in C++ Standard Template Library	**10 - 100**
El Camino College CS1 – CS2 projects	**100 - 2000**
Capstone or other projects in Computer Science senior year classes	**2000 – 10,000**
Linux command line utilities	**10,000 – 100,000**
Linux g++ compiler	**100,000 – 1,000,000**
Mozilla Firefox Browser	**1,000,000 – 10,000,000**
Microsoft windows 2000 Kernel	**10,000,000 – 100,000,000**
Debian Linux Operating System	**100,000,000 – 1,000,000,000** **Hundred million - billion**
Human Gnome	**Up to 3,300,000,000,000** **3.3 trillion**

So far you have written programs such that all code is written in main function. Certainly, no one can even fathom writing 3.3 trillion lines of code that human genome requires written in the main function, and still hope such software to work even once! Thus, in large pieces of software, software or code organization becomes a necessity. This is where we make a switch from typical computer science to the discipline of software engineering.

Modern software engineering has many ways for code organization, so that writing, debugging, testing, and maintaining software becomes easier and cost effective. First technique for code organization that appeared in FORTRAN programming language, where it proposed to break down large piece of code into modules, such that each module accomplishes one goal (among many) in the software. FORTRAN called these modules as routines or sub-routines. C++ calls modules by name "functions".

In this assignment you make your first stride in this important code organization technology, where software is divided into user-defined functions and then the main function calls them in order to operate the entire software. In this project, we have done the design for you and we ask you to code various functions. By the time you complete your senior year capstone course in computer science or software engineering, you will become expert in doing this and more complex design, using other code organization technologies, such as use of object-oriented programming etc.

Program Description

Temporary or hourly workers get paid by hours worked. Labor department laws regarding temporary help are below:

If hours worked are up to 40 hours or less then workers salary is hours worked multiplied by hourly pay rate. However, if hours worked exceeds 40 then hours above 40 must be paid by the rate of 1.5 times of regular hourly rate.

In addition, employers can withhold social security tax at federally mandated rates. Employers may have medical insurance for temporary workers for which they may pay partially.

Description of Software to Be Written

No Global Variables are allowed. Use of even a single variable or goto is an automatic zero in any assignment. All constants MUST be declared globally before function prototypes and after the includes. On top of your .cpp file there must be an ID block.

This program/software takes user input for:

1. Employee First Name
2. Employee Last Name
3. Employee hours Worked
4. Employee hourly pay rate
5. Whether they wish to enroll in health plan or not?

Software uses the constants described in background portion of this document. Additional constants are below:

- After calculating employee gross salary using labor laws from background portion, a flat 7.1% is withheld as social security tax. Thus, employee's net salary would be gross salary minus the 7.1% tax on gross salary.

- $200 is deducted before net salary computations if employee wishes to enroll in health plan. If employee's salary after 7.1% tax reduction is less than $200 then their request for enrolling in health plan is declined.

Software outputs the following information.

Greeting the user pleasantly and explaining what software does.
Employee Full Name
Number of Hours Worked
Hourly Pay Rate
Gross Salary
Tax Withheld
Whether Employee enrolled in health plan or not or whether enrollment was denied?
Cost of health plan if employee was enrolled.
Employee Net Salary
Pleasant goodbye to user, and thanking them for using software.

This software is modularized in functions. The suggested function prototypes and their description is below. Description is just a best effort narrative to be supplemented by your clarifying questions. All questions to describe the goal of a function will be answered. It is software writer's responsibility to write algorithm/pseudo code for a function and write source code for it.
Below are function proto-types and function description. When working with functions, IOAA must be done for each function. There is space provided for you to do that in this document.
Caution: Please DO NOT copy and paste function proto-types from this document into Visual Studio! You may end up with some strange problems. All proto-types must be typed and rechecked few times to ascertain the typing accuracy. Also, you should test correct functioning of each function right after it has been coding. The testing can be done in the temporary main function by simulating the call to the function.
//---
Function #1:
void greeting();
Function greeting pleasantly greets the user, explains the purpose of software/program, inputs that may be needed, and approximate time requirement from them. **This function is not allowed to take any user input from keyboard or file.**

Function #2:
double getUserData(double & payRate, string & firstName, string & lastName, bool & healthPlan);
Function getUserData gets data required from the user. Since user is required to provide five pieces of data, and a function can only return one value, the other four values are

returned by reference (Notice the ampersand (&) symbol between data type and parameter name in the function argument list. Description of function arguments is below:

payRate: Hourly pay rate of Employee. For example an hourly wage of $15 and 23 cents will be entered by the user as 15.23. The caller function has no value assigned to payRate as it is populated by the getUserData function.

firstName: One word first name of the Employee. The caller function has no value assigned to firstName as it is populated by the getUserData function.

lastName: One word last name of the Employee. The caller function has no value assigned to lastName as it is populated by the getUserData function.

healthPlan: Employee's selection or declination of health plan. Employee can enter zero to decline the health plan or 1 to enroll in it. [Recall 0 is false in C++ and 1 is true]. The caller function has no value assigned to healthPlan as it is populated by the getUserData function.

The function gets hours worked and returns it by return statement or by return mechanism.

Function #3:
double getGrossSalary(double hoursWorked, double payRate);
Function getGrossSalary gets two arguments, hoursWorked, and payRate. Function computes the gross salary as per labor laws described in background section and returns the gross salary by return statement mechanism.
Notice that parameters passed to this function are passed by value only! **This function is not allowed to take any user input from keyboard or file. Neither is function is allowed to make any outputs to console or file.**

Function #4:
double getTaxWitheld(double GrossSalary);
The function getTaxWitheld takes GrossSalary parameter by value and returns the tax withheld by return statement mechanism.
Notice that parameter passed to this function is passed by value only! **This function is not allowed to take any user input from keyboard or file. Neither is function is allowed to make any outputs to console or file.**

Function #5:
double getNetSalary(double grossSalary, double taxWitheld, bool HealthPlan, string & HealthPlanStatusMessage);

The function getNetSalary computes the net salary of the employee. The function deducts the taxWitheld from the grossSalary. Then function checks the value of Boolean parameter HealthPlan and the value of net salary. If Boolean parameter, HealthPlan is true and net salary after tax withheld is greater than or equal to $200 for health premium then health premium is deducted from net salary computed in previous step, to compute the final net salary. Depending upon the following situations the net salary is computed correctly and HealthPlanStatusMessage is set accordingly.

1. User opted for health plan and salary after tax deduction is enough to enroll in the health plan, in which case the enrollment is accepted and HealthPlanStatusMessage is set accordingly.
2. User opted for health plan but salary after tax deduction is not enough to enroll in the health plan, in which case the enrollment is denied and HealthPlanStatusMessage is set accordingly.
3. User did not opt for health plan. Then HealthPlanStatusMessage is set accordingly.

   Function is NEVER allowed to return a negative salary, as that would be a violation of business rule that no employee can be given negative salary. This requirement must be met at the level of this function and every other function.

This function is not allowed to take any user input from keyboard or file. Neither is function is allowed to make any outputs to console or file.

Function #6:
void printResults(const string & firstName, const string & lastName, double hoursWorked, double payRate, double grossSalary, double taxWitheld, double netSalary,
const string & HealthPlanStatus);
printResults function takes eight arguments and does output as mandated on output specifications listed on pages 2 and 3. The test run from program I did, show in the end of this document, one model of output. Further beautifying that model would be welcome, though not required.
Notice that except string data types, (that are passed as const reference), all other parameters passed to this function are passed by value only! **This function is not allowed to take any user input from console or from file.**

Function #7:
void goodbye();

This function says pleasant good bye to the user.

///

Function #8:
main Function:

Main function calls all the seven user defined functions described above using the algorithm that satisfies software requirements.

You must test the whole program with variation of enough data sets that outputs from them shows that all lines of code have been executed. Without using a wide enough data sets to test program the bugs may not be revealed. Test runs show some examples of inputted data that may help you widen your test data set to test your program.

///

Test Runs From my Program are shown in table below. Your program is NOT submission ready until it reproduces exactly the output shown below by five test runs.

Output Example 1

```
-----------------------------------------------------------------------------------------------
Welcome to salary calculation program.
In this program we will ask for the information that would allow us to process your pay check.
We will collect information about hours worked, hourly pay rate and medical options.
Processing will be completed in just few minutes.
-----------------------------------------------------------------------------------------------
Please enter your first name [One word only]: Satish
Please enter your last name [One word only]: Singhal
Please enter hours worked (positive number only) :40
Please enter your hourly pay rate. [For example, enter 15.23 if your
hourly pay rate is $15 and 23 cents]
20.0
Enter 1 to purchase health plan or 0 (zero) to decline: 1
---------------------------------------------------------------------------
Here are the Employee Payroll details.
Name: Satish Singhal
Hours worked: 40.00 hours
Hourly Pay Rate: $20.00
Gross Salary: $800.00
Tax withheld: $56.80
Employee opted for health insurance and was successfully registered.
Employee net salary: $543.20
```

Output Example 2 is below

```
Welcome to salary calculation program.
In this program we will ask for the information that would allow us to process your pay check.
We will collect information about hours worked, hourly pay rate and medical options.
Processing will be completed in just few minutes.
-------------------------------------------------------------------------------------------
Please enter your first name [One word only]: Jim
Please enter your last name [One word only]: Jones
Please enter hours worked (positive number only) :10
Please enter your hourly pay rate. [For example enter 15.23 if your
hourly pay rate is $15 and 23 cents]
10.1
Enter 1 to purchase health plan or 0 (zero) to decline: 1
-----------------------------------------------------------------------
Here are the Employee Payroll details.
Name: Jim Jones
Hours worked: 10.00 hours
Hourly Pay Rate: $10.10
Gross Salary: $101.00
Tax withheld: $7.17
Employee opted for health insurance but was declined because of insufficient salary.
Employee net salary: $93.83
-------------------------------------------------------------------------------
Thank you for using our program.
Please have a pleasant day.
```

Output Example 3 is below

```
-------------------------------------------------------------------------------------------
Welcome to salary calculation program.
In this program we will ask for the information that would allow us to process your pay check.
We will collect information about hours worked, hourly pay rate and medical options.
Processing will be completed in just few minutes.
-------------------------------------------------------------------------------------------
Please enter your first name [One word only]: Jerry
Please enter your last name [One word only]: Jones
Please enter hours worked (positive number only) :10
Please enter your hourly pay rate. [For example enter 15.23 if your
hourly pay rate is $15 and 23 cents]
15.23
Enter 1 to purchase health plan or 0 (zero) to decline: 0
-----------------------------------------------------------------------
Here are the Employee Payroll details.
Name: Jerry Jones
Hours worked: 10.00 hours
Hourly Pay Rate: $15.23
Gross Salary: $152.30
Tax withheld: $10.81
Employee did not opt to register in health plan.
Employee net salary: $141.49
-------------------------------------------------------------------------------
Thank you for using our program.
Please have a pleasant day.
```

Output Example 4 is below

```
--------------------------------------------------------------------------------------------
Welcome to salary calculation program.
In this program we will ask for the information that would allow us to process your pay check.
We will collect information about hours worked, hourly pay rate and medical options.
Processing will be completed in just few minutes.
--------------------------------------------------------------------------------------------
Please enter your first name [One word only]: Tom
Please enter your last name [One word only]: Rogers
Please enter hours worked (positive number only) :50
Please enter your hourly pay rate. [For example enter 15.23 if your
hourly pay rate is $15 and 23 cents]
35.51
Enter 1 to purchase health plan or 0 (zero) to decline: 1
------------------------------------------------------------------------
Here are the Employee Payroll details.
Name: Tom Rogers
Hours worked: 50.00 hours
Hourly Pay Rate: $35.51
Gross Salary: $1953.05
Tax withheld: $138.67
Employee opted for health insurance and was successfully registered.
Employee net salary: $1614.38
---------------------------------------------------------------------------------
Thank you for using our program.
Please have a pleasant day.
```

Output Example 5 is below

```
--------------------------------------------------------------------------------------------
Welcome to salary calculation program.
In this program we will ask for the information that would allow us to process your pay check.
We will collect information about hours worked, hourly pay rate and medical options.
Processing will be completed in just few minutes.
--------------------------------------------------------------------------------------------
Please enter your first name [One word only]: Lisa
Please enter your last name [One word only]: Adams
Please enter hours worked (positive number only) :22
Please enter your hourly pay rate. [For example enter 15.23 if your
hourly pay rate is $15 and 23 cents]
12.0
Enter 1 to purchase health plan or 0 (zero) to decline: 1
------------------------------------------------------------------------
Here are the Employee Payroll details.
Name: Lisa Adams
Hours worked: 22.00 hours
Hourly Pay Rate: $12.00
Gross Salary: $264.00
Tax withheld: $18.74
Employee opted for health insurance and was successfully registered.
Employee net salary: $45.26
---------------------------------------------------------------------------------
Thank you for using our program.
```

```
Please have a pleasant day.
```