Swinburne University of Technology Sarawak

COS10009 Introduction to Programming

Standard Input / Output, Variable (Lab 01)

Pass Task 1.1 : Hello World

Task : Create your own Hello World program using the command line interpreter. This will help ensure that you have all of the software installed correctly, and are ready to move on with creating other programs.

To Do :
The first task includes the steps needed for you to install the tools you will need in this unit. You will then use these tools to create the classic 'Hello World' program.

1. Install the tools you need to get started for your operating system. Install Atom, SublimeText or Notepad++ or equivalent (or use a text editor of your choice). Preferably install VS Code. See the installation notes on Lecture Slides.

2. Open your text editor, and create a new file.

3. Enter a C program to print Hello World. It should appear as below:

```
#include <stdio.h>

void main(){
    printf("Hello World\n");
}
```

4. Save the file as hello.c in your code directory.

5. Open a Terminal (or a CMD shell in Windows), then perform the following commands:

- Change into the directory containing your code using the cd command.
- List the files in this directory using the ls command
- Print the working directory using the pwd command
- Run your program by typing gcc hello.c -o hello.exe
- Run hello.exe

Tip: Bash commands (e.g., cd, ls, pwd) do not like spaces in directory or file names (e.g., My Documents, or hello.c). If you have a space in the name of something you need to add in a reverse slash: My\ Documents and hello_world.c Avoid spaces in the names of your files and folders!

Pass Task 1.2 Reading and Writing (Hello User)

Task: Create program that reads input from user and display output according to the input from the user. This program will allow you to demonstrate simple reading and writing from the terminal window and using basic data types in C.

To Do: Variable you are telling the computer to create a variable for use within the function or procedure. You must give the variable a name and keep in mind the type of data it will store. The computer will then set aside space for you to store a value for that variable. You can then write values to the variable and read values back.

To explore this topic, we will create a Terminal program that will:

- ask the user to enter their name, age, and weight in metres,
- calculate and output the year the user was born, and
- calculate and output the user's height in inches

1. Declare the Hello User program using the program start code shown below (available in the Resources for this task)

```
#include <stdio.h>
#include "./functions.h"
#define INCHES 0.393701 // This is a global constant

void main()
{
    char name[256];
    char family_name[256];
    int year_born;
    printf("What is your name ?\n");
    /* Insert scanf here */
    printf("Your name is %s !\n", name);

    printf("What is your family name ?\n");
    /* Insert scanf here */
    printf("Your family name is %s !\n", family_name);

    printf("What year were you born ? \n");
    /* Insert scanf here */

    // Calculate age
    int age = get_age(year_born);
    printf("So you are %d years old\n", age);

    // Calculate height in inches
    float height_cm;
    printf("Enter your height in cms (i.e as a float) \n");
    /* Insert scanf here */
    printf("Your height in inches is: %0.2f\n", height_cm * INCHES );
   printf("Finihsed\n");


}
```

Tip: This code is the basic program template that you can use whenever you create a new program. The Main procedure will have the program's main instructions.

2. Extend the code in main so that it does the following:

a. Read the user's name (a String, prompt with 'Please enter your name: ') and store it in the name variable.

b. Print out the user's name followed by an exclamation mark.

c. Read the user's family name, and store it in the family_name variable.

d. Print out the user's family_name followed by an exclamationmark.

e. Read in the user's year of birth.

f. Convert the year of birth to an integer then calculate the user's age and print it out.

g. Prompt for and read in the user's height in metres.

h. Convert the height to inches and print out the result.

Pass Task 1.3 Desk Check

Task: This task will help you to learn about variables and to desk check and test a sequence of statements using variables.

To Do:

1. Use desk checking to demonstrate that a program works as expected (Program 2 below).
2. See the example (Program 1) below and follow that format.
3. Complete the for Pass Task 1.3 answer sheet.
4. Run the C code provided for Program 2 (in the Task's Resources) and check that the test results are what is expected.

Program 1: Add 2 numbers (Example of Desk Checking)

Required Variables:
Integer: a, b, c.

Pseudocode:
Read the value of a
Read the value of b
Add a to b and assign the result to c
Print the value of c to the terminal.

Test Data:

Variable	First Data Set	Second Data Set
a	10	3
b	5	4

Expected Result:

First Data Set	Second Data Set
15	7

Desk Check:

	Statement	**a**	**b**	**c**	**output**
First Pass	Read the value of **a**	10			
	Read the value of **b**		5		
	Add **a** to **b** and assign the result to **c**			15	
	Print the value of **c** to the terminal				15
Second Pass	Read the value of **a**	3			
	Read the value of **b**		4		
	Add **a** to **b** and assign the result to **c**			7	
	Print the value of **c** to the terminal				7

Program 2: Calculate Meal Total *(You do this one)*

Required Variables:
Integer: appetizer_price, main_price, dessert_price
Real (floating point): total_price

Pseudocode:
Read the value of **appetizer_price** (in cents)
Read the value of **main_price** (in cents)
Read the value of **dessert_price** (in cents)
total_price = appetizer_price + main_price + dessert_price
total_price = total_price / 100 *#Comment: convert to dollars*
Print '$' then the value of **total_price** showing two decimal places.

Test Data:

Variable	First Data Set	Second Data Set
appetizer_price	1030	1240
main_price	3400	4100
dessert_price	850	980

Expected Result:

First Data Set	Second Data Set
$52.80	$63.20

Complete the following desk checking on the answer sheet provided:

	Statement	*appetizer_price*	*main_price*	*dessert_price*	*total_price*	*output*
First Pass	***Read the value of*** *appetizer_price*					
	Read the value of *main_price*					
	Read the value of *dessert_price*					
	Calculate the *total_price*					
	Convert to dollars					
	Output the *total_price*					
Second Pass	***Read the value of*** *appetizer_price*					
	Read the value of *main_price*					
	Read the value of *dessert_price*					
	Calculate the *total_price*					
	Convert to dollars					
	Output the *total_price*					

Credit Task 1.1 Interest Calculator

With compound interest, interest is paid on the principle and any interest received during the period of the investment. For example, if you invest $1000 at 15% interest then in the first year you will receive $150 interest. This interest is then added to your investment, and in the second year you will receive $172.50 interest (15% of the $1150). The following formula can be used to calculate the amount of interest an investment will receive over a period of time.

$$I = P (1 + i)^n — P$$

For example, after 5 years our $1000 invested at 15% will have accrued $1011.375 in interest.

$$I = P (1 + i)^n — P$$
$$I = 1000 * (1 + 0.15)^5 — 1000$$
$$I = 1011.357$$

1. Create a program named InvestmentCal, which will read and write values from the consol.
2. Declare variables for principle, years and rate.
3. Get the user to enter a principle, rate and duration, then the program will output the interest earned from the investment.

Hint: The user is expected to enter the interest rate in percentage (so 15 for 15%)