

CS1010E Programming Methodology
Semester 1 2016/2017

Week of 22 August – 26 August 2016

Tutorial 1 Suggested Answers

Basics of C Programming with Numerical Computations

1. The *Edit–Compile–Run* cycle in program development is second nature to all programmers. In essence, we use an editor (**vim** or **nano**) to type out the source code/program, which is then compiled via a compiler (**gcc**) to machine executable instructions which is, in turn, executed/run. Experience the *Edit–Compile–Run* cycle by typing out the following program, compiling and executing it. Take note of any errors and rectify the program accordingly.

```
#include <stdio.h>

int main(void) {
    int cur, prev1=1, prev2=1;

    cur = prev1 + prev2;
    prev2 = prev1;
    prev1 = cur;

    cur = prev1 + prev2;
    prev2 = prev1;
    prev1 = cur;

    cur = prev1 + prev2;
    prev2 = prev1;
    prev1 = cur;

    printf("cur is %d; prev1 is %d; prev2 is %d\n", cur, prev1, prev2);

    return 0;
}
```

2. Being able to trace through the execution of a program without going through the *Edit–Compile–Run* cycle is known as *program tracing*. Program tracing involves building up a *mental model* of program execution in your mind. By tracing a program, you will gain a stronger understanding of the program execution flow and improve your programming skill. Trace the execution of the program in question 1, paying particular attention to variable declarations and changes to the values of the variables. Verify your program trace by including your own *debugging printf* statements at strategic points in your program in question 1 to track the value changes of variables.

*Refer to the slideshow **mentalModel.pps**.*

3. The assignment statement in C allows variables to take on different values at different points of time during the execution of a program. You are given the following program that prints out the values of two variables, `num1` and `num2`.

```
#include <stdio.h>

int main(void) {
    int num1=10, num2=20;

    /* include statements to swap num1 and num2 */

    printf("num1 is %d; num2 is %d.\n", num1, num2);

    return 0;
}
```

- (a) Include further assignment statements to swap the values of `num1` and `num2` such that the output becomes

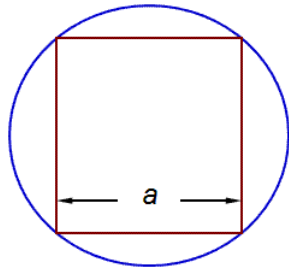
`num1 is 20; num2 is 10.`

You are not allowed to change the initialized declarations of `num1` and `num2` or modify the `printf` statement. However, you may add one additional variable.

- (b) Replace the initial assignments of `num1` and `num2` so that the values are now read from the user. Test the *generality* of the program and observe if the values are indeed swapped.

<pre>#include <stdio.h> int main(void) { int num1=10, num2=20, temp=0; temp = num1; num1 = num2; num2 = temp; printf("num1 is %d; ", num1); printf("num2 is %d.\n", num2); return 0; }</pre>	<pre>#include <stdio.h> int main(void) { int num1=0, num2=0, temp=0; printf("Enter two numbers: "); scanf("%d%d", &num1, &num2); temp = num1; num1 = num2; num2 = temp; printf("num1 is %d; ", num1); printf("num2 is %d.\n", num2); return 0; }</pre>
--	---

4. A square is inscribed in a circle with all the four corners touching the circumference of the circle. An algorithm to calculate the area of the circle given the side of an inscribed square is described below.



Step 1. Read in the side of the square, a .

Step 2. $radius \leftarrow \sqrt{2 \times \left(\frac{a}{2}\right)^2}$.

Step 3. $area \leftarrow \pi \times radius^2$.

Step 4. Print the $area$.

Translate the algorithm into a C program by defining the `main` function. Declare all variables of `double` type, and use 3.14159 as the value of π . The following is a sample run of the program. User input is underlined.

```
Enter side of the square: 5.65
Area of the circle is 50.143703
```

Hint: There is no need to perform the square root operation.

```
/*
   This program computes the area of the circle given
   the side of an inscribed square as input.
*/
#include <stdio.h>

#define PI 3.14159

int main(void) {
    double side, radiusSq, area;

    printf("Enter side of the square: ");
    scanf("%lf", &side);

    radiusSq = 2 * (side/2) * (side/2);
    area = PI * radiusSq;

    printf("The area of the circle is %f\n", area);

    return 0;
}
```