

# CMPE 108 - Experiment 3

## Selective Structures

### OBJECTIVES:

- Understand how to edit, compile and execute C computer codes.
- Understand C programming: sequential and selective structures

### NOTES:

- You should prepare the preliminary work before coming to the laboratory session and bring soft or hard copies of the preliminary work with you.
- Before writing a computer code, you should do the following steps:
  - 1) understand and analyze the problem,
  - 2) develop an algorithm and/or flowchart,
  - 3) convert the algorithm and/or the flowchart into a C code.

### PRELIMINARY WORK:

#### 1. What is the output of the following C code fragments?

**a)**

```
int a=9, b=4;
float c,d;
c = a/b;
d = (float)a / (float)b;
printf("c=%.2f,d=%.2f",c,d);
```

**b)**

```
int i = 5;
float j = 314.15;
char cr = '$';
printf("%5i", i);
printf("%6.1f", j);
printf("%f", j);
printf("%.1e", j);
printf("%10.2e", j);
printf("%c", cr);
```

**c)**

```
int x, a = 5, b = 10;
x = a++ + b++;
printf("x=%d\n a=%d\n b=%d\n", x, a, b);
```

**d)**

```
int x, a = 5, b = 10;
x = ++a + ++b;
printf("x=%d, \n a=%d\n b=%d\n", x, a, b);
```

**e)**

```
int x, a = 5, b = 10;
x = a++ + --b;
printf("x=%d, \n a=%d\n b=%d\n", x, a, b);
```

```
f) int a = 3, b = 5;
a += b;
printf("a=%d\n b=%d\n", a, b);
```

```
g) int a = 3, b = 5;
a += b++;
printf("a=%d\n b=%d\n", a, b);
```

```
h) int a = 3, b = 5;
a *= b - (b / 2);
printf("a=%d\n b=%d\n", a, b);
```

```
i) int a=3, b=5, c=4;
a += b -= --c * 2;
printf("a=%d, \n b=%d \nc=%d\n", a, b, c);
```

```
j) float x = 4;
x += 2 / 3;
printf("x=%lf\n", x);
```

```
k) float x = 4;
x += 2.0 / 3;
printf("x=%f\n", x);
```

```
l) float x = 4;
x += (float)2 / 3;
printf("x=%f\n", x);
```

```
m) int a = 5, b = 8;
a = b;
b = a;
printf("a=%d\n b=%d\n", a, b);
```

```
n) int a = 5, b = 8, temp;
temp = a;
a = b;
b = temp;
printf("a=%d\n b=%d\n", a, b);
```

**2. What is the out of the following C code fragments for the following declarations?**

```
float x;
int i, j, k;.
```

```
a) i=5; j=2;
if(i=50||j==2)
printf("%d", i-j);
```

```
b) i=5; j=2;
if(i==50||j==2)
printf("%d", i-j);
```

**c)** `i=1;j=1;k=1;  
k+--i++ + ++j;  
printf("%d %d %d\n", i,j,k);`

**d)** `i=1;j=2;x=3.0;  
x/=j=++i*2.5;  
printf("%f %d %d\n",x,i,j);`

**e)** `i=1;j=2;x=3.0;  
x/=j=i++*2.5;  
printf("%f %d %d\n",x,i,j);`

**f)** `i=-1; j=3;  
k = ++i&&--j;  
printf("i = %d j = %d k = %d\n",  
i, j, k);`

**g)** `i=-1; j=3;  
k = i++&&--j;  
printf("i = %d j = %d k = %d\n",  
i, j, k);`

**h)** `i=-1; j=0;  
k = i++||--j;  
printf("i = %d j = %d k = %d\n",  
i, j, k);`

**i)** `i=6;j=1;k=3;  
k=i<j<k;  
printf("i = %d j = %d k = %d\n",  
i, j, k);`

**j)** `j=9;k=2;  
i=j%k?++j:++k;  
printf("j = %d k = %d\n", j, k);`

**k)** `i=1, j=3;k=7;  
i = (i < j)% j  
printf("i = %d j = %d k = %d\n",  
i, j, k);`

**l)** `i=1, j=5;k=7;  
i--i && --j || --k  
printf("i = %d j = %d k = %d\n",  
i, j, k);`

**m)** `i=1, j=5;k=7;  
k= (++i +4)/(--k - j--)  
printf("i = %d j = %d k = %d\n",  
i, j, k);`

3. Write separate C programs to evaluate the given program fragments and write the produced outputs in the corresponding boxes. Use the following declarations:

```
int a, b, c, sum;
a=3, b=5, c=4, sum=0;
```

a)

```
if (b%a)
{
    if (a>b)
        printf("a is greater");
    else
        printf("b is greater");
}
```

b)

```
if (b%a)
    (a>b) ? printf ("greater a") : printf("greater b");
```

c)

```
(a>b)?sum=a+b:(a<c)?sum=a+c:sum=a+b+c;
printf("sum is %d\n", sum);
```

d)

```
if (a>b)
{
    sum= a+b;
    printf("sum is %d\n", sum);
}
else if (a<c)
{
    sum= a+c;
    printf("sum is %d\n", sum);
}
else
{
    sum= a+b+c;
    printf("sum is %d\n", sum);
}
```

4. Rewrite the following using if-else-if structure.

```
switch(i) {
case 0: k+=i;break;
case 1: k*=i;break;
case 2: case 3: k/=i;break;
default: k%=i;
}
```

## TASKS during the LAB hours:

1. Write an `if` statement that computes and prints the circumference or the area of a square using the formulas:

```
circumference = 4* side
area = side * side
```

Enter the value of integer variable `side` from the keyboard. Assume that if `user_request` is 1, the program finds the circumference and prints and if it is 2, the program finds the area and prints.

2. Write a program for the following problem. An instructor needs a program that accepts student identification number `std_id` and three exam grades, `exam1`, `exam2` and `final_exam` for a course, as input and then determines and outputs for the student, the semester average and the final letter grade of the course according to the following table:

Semester Average	Letter Grade
90-100	A
80-89	B
70-79	C
60-69	D
0-59	F

The semester average for the student is computed using the following formula:

```
semester_average = 0.20 * exam1 + 0.30 * exam2 + 0.50 * final_exam
```

Student identification numbers are integers but not 0 or negative numbers. The program should print an appropriate message when 0 or negative number is entered for the student identification.

A session of the program should have the following appearance:

```
Enter student id_no: 1100
Enter exam grade 1: 70
Enter exam grade 2: 80
Enter final exam grade: 100
Semester average for student 1100: 88
Letter grade for student 1100 : B
```

If 0 or negative number is entered for student identification:

```
Enter student id_no: 0
Student id_no is wrong.
```

3. Consider the following quadratic equation:

$$A \cdot X^2 + B \cdot X + C$$

Then, the roots of this equation can be obtained as

$$X1 = (-B + \text{SQRT}(\text{Disc})) / (2 \cdot A)$$

$$X2 = (-B - \text{SQRT}(\text{Disc})) / (2 * A)$$

where the discriminator (**Disc**) is given by

$$\text{Disc} = B^2 - 4 * A * C$$

- a) Write a C program that will read the value of the inputs A, B, and C, and then check for the following conditions and do the corresponding tasks:
- If **Disc > 0**, then find the roots **X1** and **X2** and then display the result as,  
**The roots are different: X1=..... X2=.....**
  - If **Disc = 0**, then find the roots **X1** and **X2** and then display the results as  
**The roots are equal: X1 = X2=.....**
  - If **Disc < 0**, then display the following message:  
**The roots are imaginary.**
- b) Re-write the same code by using switch statement rather than if/else.