ENGG1003 - PASS Session 2
Mitchell Deltoer

Monday	**14-15**	**ES238**
Wednesday	**12-13**	**ES238**
Thursday	**10-11**	**MCLG42**

More Arithmetic in C

1. What is the meaning of the *modulus* operation in C?

2. Evaluate the following C expressions. You can confirm your answers by printing the result on your computer.
   HINT: Be careful of integer division.

   (a) `12 / 5;`

   (b) `12 % 5;`

   (c) `12.0 / 5;`

   (d) `12 % 2;`

   (e) `5 / 12;`

   (f) `5 % 12;`

3. Match the following relational and boolean operations with their C counterparts.

Operator	Meaning
<	Greater than
>	Equal to
<=	Or
>=	Not equal to
==	Not
!	And
!=	Less than
||	Greater than or equal to
&&	Less than or equal to

4. Complete the following table for **OR** and **AND** logic.

x	y	x OR y	x AND y
1 (True)	1 (True)		
1 (True)	0 (False)		
0 (False)	1 (True)		
0 (False)	0 (False)		

5. What is the value of `int ans` after the following relational operations? Note the following variable initialisations.
   HINT: The result of a relational operation is 0 or 1. C treats 0 as Boolean FALSE and any non-zero as TRUE.

   `int x = 0, y = 23, z = -2; float num = 1.50;`

   (a) `ans = y > x;`

   (b) `ans = !(z > x);`

   (c) `ans = x <= z+2;`

   (d) `ans = x || y;`

   (e) `ans = x && 4*z;`

   (f) `ans = (num <  z) && x;`

   (g) `ans = (num < = z) == x;`

   (h) `ans = !z;`

   (i) `ans = !x && y;`

   (j) `ans = !(x <= y) || (z*y <= z-y);`

IF statements

`IF` statements allow us to choose between multiple different code segments, depending one or more *conditions*.

6. Given the following code listing, what will be displayed for the user for each set of x and y?

     (a) `int x = 0, y = 12;`   (b) `int x = 32, y = 0;`

     (c) `int x = -23, y = 12;`   (d) `int x = 1, y = 10;`

```c
if(x <= 0){
    printf("Flood levels reached.\n");
} else if(!y && x){
    printf("Flood gates obstructed!\n");
} else{
    printf("Error - system check required.\n");
}
```

WHILE loops

Loops are an important construct in C that allow us to execute code segments multiple times or even indefinitely. It is important to make sure that the loop condition is updated each loop or you could get stuck in an endless loop.

7. Examine the following code listing and determine what will be the output. Have a go first before you confirm the result on your computer.

```c
int x = 3, y = 7;
while(x < y){
    printf("%d", x);
    x++;
}
```

Practice Programming

8. Write code for a C program that will calculate and display the length of the hypotenuse of a right-angled triangle, given the other two side lengths as input. (**HINT:** I hope you have seen Pythagoras' Theorem before).

9. Write code for a program that takes in a number from the user. The program should then tell the user if that number is between 0 and 10.

10. Write code to add all the integers up until the number the user has input, then display the result.

11. Write code for a program that takes in two numbers from the user, then displays the largest of the two numbers.

12. Write code that reads in a dollar amount (integer) and breaks it into smallest possible number of bank notes ($100, $50, $20, $10, $5) and coins ($2, $1). The results show be (neatly) displayed to the user.

13. Write code for a program that takes in a student's mark for a course they completed last semester. The program should then display if the user got a HD, D, C, P or F for that course.