Data Structures and Program Design Using C
SCS 1301
AY 22 - Semester 1

Exercise III

Dr Manjusri Ishwara

1. For the following exercises, use variants of **if-else** only:

   (a) Write a program to check whether a given integer value is even.

   (b) Extend the program in (a) above to check for odd integer values.

   (c) Write a program to display if a given character is an upper or lower case letter.

   (d) Extend the program in (c) above to recognize the case of the character and convert it to the opposite case. E.g., Upper case to lower case and vice versa.

   (e) Extend the program in (d) above to output characters other than letters as "numbers" where it is a number and "Other Characters" for all other possibilities.

2. For the following exercises, use variants of **switch-case** only:

   (a) Write a program to perform mathematical operations on two integer variables where the operator is given as a character variable. Consider addition, subtraction, multiplication, modulo, and division operators.

   (b) Write a program that converts an integer from zero (0) to nine hundred and ninety-nine (999) given as a variable to words corresponding to the English pronunciation. Examples are given below.

   - 0 => zero
   - 98 => Ninety Eight
   - 711 => Seven Hundred and Eleven

3. For the following exercises, use **while** and **do-while** loops.

   (a) Write a program to print all natural numbers from 1 to 50 using a while loop.

   (b) Write a program to display the '$' symbol 32 times using a while loop.

   (c) Write a program to display all integers from 10 to 0 in reverse order using a while loop.

   (d) Write a program to display all floating point numbers from 10.0 to -5.0 in 0.1 decrements.

   (e) Write a program to display all counter numbers from 10 to 99. Indicate whether they are multiples of 2, 3, or 5. The output should match exactly as follows:

```
10        -multiple of 2        multiple of 5
11
12        -multiple of 2        multiple of 3
...
99        -multiple of 3
```

(f) Write a program to display all simple letters in the English alphabet.

4. For the following exercises, use the **for loop**.

   (a) Write a program to display all integers from 20 to 35.

   (b) Write a program to display the following pattern. Each loop should print one '*' per iteration.

   ****

   (c) Write programs to display the patterns below.

   i.   *
               *
               *
               *
               *

   ii.   ********
               ****
               ****
               ****
               ****

   iii.   *
               **
               ***
               ****
               *****

   iv.   *
               **
               ***
               ****
               *****
               ****
               ***
               **
               *

   v.   *
               ***
             *****
           ****
         ****
           ****
             *****
               ***
                 *

```
vi.     *       1
        **      3
        ***     6
        ****    10
        *****   15
        ****
        ***
        **
        *


vii.    ********
         ********
          ********
           ********
            ********
```

5. What would be the output for each print statement in the code given below? You must dry run first and then execute the code to verify the results.

```c
#include <stdio.h>
int main()
{
    double d;
    float f;
    long l;
    int i;

    i = l = f = d = 100/3;
    printf("%d\t%l\t%f\t%f\n",i,l,f,d);

    d = f = l = i = 100/3;
    printf("%d\t%l\t%f\t%f\n",i,l,f,d);

    i = l = f = d = 100/3;
    printf("%d\t%l\t%f\t%f\n",i,l,f,d);

    d = f = l = i = (float)100/3;
    printf("%d\t%l\t%f\t%f\n",i,l,f,d);

    i = l = f = d = (double)(100000/3);
    printf("%d\t%l\t%f\t%f\n",i,l,f,d);

    d = f = l = i = 100000/3;
    printf("%d\t%l\t%f\t%f\n",i,l,f,d);

    return 0;
}
```

6. What would be the output for each print statement in the code given below? You must dry run first and then execute the code to verify the results.

```c
#include <stdio.h>

int main()
{
    double d = 3.2, x;
    int i =  2, y;

    x = (y = d/i)*2;
    printf("%f\t%d\n", x, y);

    y = (x = d/i)*2;
    printf("%f\t%d\n", x, y);

    y = d * (x=2.5/d);
    prinf("%f\n",y);

    x = d * (y = ((int)2.9+1.1)/d);
    printf("%f\t%d\n", x, y);
}
```
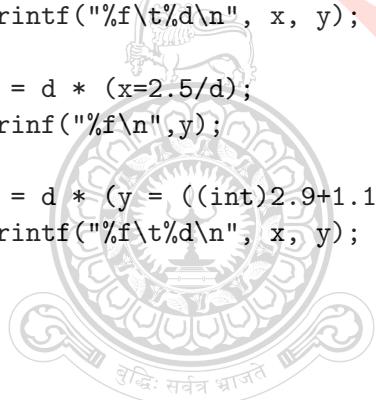
*****

4