KSU/CCIS/CS	**CSC 215**	**Midterm Exam 2- Spring 14** **1:30 Hours**
Name: ……………………………………………………..	**ID: ……………………………………………………..**	

EXERCISE 1 **(/20pts)**

Write True/ False

Calling **free** on the same address more than once is ok.	
strlen(s) returns the number of characters in s including the terminating character.	
free can be used to release **any** allocated memory back to the operating system.	
When using stderr, the output appears on screen even if stdout is redirected.	
A structure type definition may be local to a function or it may be external to any function.	
A pointer of one type can be converted to a pointer of another type by using an explicit cast.	
strdup copies a string into a newly created location	
The value returned by **isalnum('9')** is 0	
Two member variables in different structures can have the same name.	
Both **malloc** and **calloc** initialize allocated memory to null	

EXERCISE 2 **(/20pts)**

Select the correct answer

1. Which of the following is **NOT** an error when using free:
   a) Use free on a NULL pointer
   b) Use free on a pointer that has already been freed
   c) Use free on a memory address directly returned by malloc.
   d) Use free on a memory address that has been statically allocated

2. Which of the following cannot be a structure member?
   a) Another structure
   b) Function
   c) Array
   d) None of the mentioned

3. If **b** is a pointer to a structure, which of the following accesses its member variable **var**.
   a) b->var;
   b) b.var;
   c) b-var;
   d) b>var;

4. Which of the following is the proper keyword or function to allocate memory in C?
   a) new
   b) malloc
   c) create
   d) allocate

5. Which of the following is the proper keyword or function to deallocate memory?
   a) free
   b) delete
   c) clear
   d) dealocate

EXERCISE 3 **(/20pts)**

1. Write the output of the following C program

```
#include <stdio.h>
int main()
{
  char s[100] = "riyadh";
  char *p1 = &s[0];
  printf("The value of *p1 is %c\n", *p1);
  char *p2 = &s[4];
  printf("The value of p2-p1 is %d\n", p2 - p1);
  return 0;
}
```

2. Write the output of the corresponding segment of code

```
const char haystack[25] = "I love my csc215 course.";
const char needle[4] = "215";
char *ret;
ret = strstr(haystack, needle);
printf("The return value is: %s\n", ret);
```

3. Write the output of the corresponding segment of code

```
const char str[] = "csc215";
const char ch = 's';
char *ret;
ret = strrchr(str, ch);
printf("The return value is: %s\n", ret);
```

4. Write the output of the corresponding segment of code

```
int var1 = 'Q';
int varr = '3';

if( islower(var1) )
    printf("var1 = |%c| is lowercase character\n", var1 );
else
   printf("var1 = |%c| is not lowercase character\n", var1 );

if( islower(var2) )
    printf("var2 = |%c| is lowercase character\n", var2 );
else
   printf("var2 = |%c| is not lowercase character\n", var2 );
```

EXERCISE 4 (/20pts)

1. Write the code to create an array called **f** of 8 floats and dynamically allocate the memory to the elements of the array and initialize the values of **f** to zero.

2. Declare a structure called **person** with the following elements: name as character array of size 20, age as an integer and gender as character.

3. Declare a structure variable called student of type **person** (from previous question) with initial values: name “Ahmed”, age 20 and gender ‘M’

4. Write the function **isupper** as defined in ctype library. You may **NOT** use any other function from the ctype library.

```
int isupper(int c){









}
```

EXERCISE 5 **(/20pts)**

Write a C program that implements the following requirements:

1- A structure type called Employee that has the following member variables:

- idNum as integer,
- payRate as double,
- birthdate as structure with year month and day as integer member variables. (This structure must be nested inside Employee)
- hours as double.

2- A function called calcNet that takes a pointer the structure Employee and returns the net pay of the Employee using the formula: netpay = payRate * hours.

3- A main function that:

- Creates a pointer to a structure variable called **emp** of type Employee
- Initialized the structure pointed to by **emp** with the initial values: Id: 555, payRate: 100.0, birthdate: 1/1/2000, hours 40.
- Prints the employer's birthday.
- Calculates and print the net pay of the **emp** net pay.

BONUS (5pts)

Write a function called **tofile** that take **emp** as an argument and write the employee's information into a file called "employee.txt". If this function is called in the above main function, the content of the file should look as follows:

```
The employee's ID is: 555

The employee's payRate is: 100.00

The employee's birthdate is: 1/1/2000

The employee's hours are: 40 hours
```