**King Saud University**
**College of Computer and Information Sciences**
**Computer Science Department**

| | |
|---|---|
| **Course Code:** | CSC 215 |
| **Course Title:** | Procedural Programming |
| **Semester:** | Semester 2 of year 36/37 |
| **Exercises Cover Sheet:** | **Final Exam** |

**Duration: 120 minutes**

| | |
|---|---|
| Student Name: | |
| Student ID: | |
| Student Section No. | |

| **Computer Science B.Sc. Program: NCAAA: Intended Learning Outcomes (ILO) Student Outcomes ABET: Program Learning Outcomes (PLO) Student outcomes** | | **Question No. Relevant Is Hyperlinked** | **Covering %** |
|---|---|---|---|
| **NCAAA** | **1. Knowledge (NCAAA)**<br>**Suggested verbs (list, name, record, define, label, outline, state, describe, recall, memorize, reproduce, recognize, record, tell, write)** | Exercise 1 | 20% |
| | (i) Use current techniques, skills, and tools necessary for computing practices;<br>*The students learn how to use Integrated Development Environment to compile and run C programs. Students also learn the differences between procedural and object oriented languages* | Exercise 1 | 20% |
| **NCAAA** | **2. Cognitive Skills (NCAAA)**<br>**Suggested verbs (estimate, explain, summarize, write, compare, contrast, diagram, subdivide, differentiate, criticize, calculate, analyze, compose, develop, create, prepare, reconstruct, reorganize, summarize, explain, predict, justify, rate, evaluate, plan, design, measure, judge, justify, interpret, appraise)** | Exercises 2, 3,4 | 50% |
| **ABET** | b. An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution;<br>*Students learn how to manage memory using dynamic memory allocation based on problem requirement analysis*. | Exercise 2 | 30% |
| | c. An ability to design, implement and evaluate a computer-based system, process, component or program to meet desired goals.<br>*Students write procedural C programs.* | Exercise 3,4 | 20% |

**Exercise 1:** Write "True" in front the correct statements and "False" in front the wrong statements: ( /20 pts)

| | | |
|---|---|---|
| 1 | A pointer is a variable that stores address of other variable. | |
| 2 | `int num, c[2]={6,3};`<br>`num = *++c ;`<br>After execution, the value of **num** is 7. | |
| 3 | A variable that is known only within the function in which it is defined, is called a local variable. | |
| 4 | Three streams are automatically opened when program execution begins: the standard input, the standard output and the standard error. | |
| 5 | Arithmetic operators "+", "-", "++"and "--"can be applied to pointers. | |
| 6 | Linear search works only on arrays that are sorted. | |
| 7 | **pow (x, y)** returns the result of raising y to the power x. | |
| 8 | Binary search returns zero if the key value is not founded in a sorted array. | |
| 9 | **isalphanum(0)** returns 1. | |
| 10 | All functions in the Math.h library take double as an argument | |

**EXERCISE 2**: Select the correct answer ( /40pts)

1) The output of the **rand() % 50** is:

A. random variable from 1 to 50
B. random numbers from 0 to 49
C. random variable less than 50 including negative numbers
D. none of the above

2) The function **char *strcpy(char *dest, const char *src)** from the string.h library:

A. Copies string src to dest, stopping after the terminating null character has been moved.
B. Copies string src to dest, without the terminating null character has been moved.
C. copies up to maxlen characters from src into dest, truncating or nullpadding dest
D. None of the above

3) A pointer in C can point to:

A. A variable.
B. A function.
C. All of above.
D. None of Above

4) Evaluate !(1 && !(0 && 1)).

A. 1
B. 0
C. Run time Error
D. Compile Error

5) Which of the following functions concatenate two strings?

A. concat(…);
B. stringcat(…);
C. cat(…);
D. strcat(…);

6) What is the final value of x when the code:
int x;
for(x=0; x<20; x++) { } is run?

A. 20
B. 19
C. 0
D. 1

7) Which properly declares a variable of struct test?

A. struct test;
B. struct test var;
C. test;
D. int test;

8) Which is **NOT** correct about **char *strdup(char *s)** from the string.h library:

A. Copies a string into a newly created location.
B. makes a duplicate of string s, obtaining space with a call to malloc
C. The user is responsible for freeing the space allocated by strdup when it is no longer needed.
D. The allocated space is strlen(s) bytes long.

9) What is the output of the below code?

```
#include<stdio.h>
main()
{
  for(;;)
     printf("CSC Final");
}
```

A. Infinite loop
B. Prints “Hello” once.
C. No output
D. Compile error

10) The conversion specification %ld is used with:

A. long double argument.
B. long int argument.
C. Char argument.
D. String argument.

11) What is the output of the following program?

```
#include<stdio.h>
main()
{
  char *s = "Hello";
  while(*s!='\0')
        printf("%c", *s++);
}
```

A. Hello
B. Helloellolloloo
C. ello
D. Compile error

12) Consider the following program fragment: The correct values of a and c are

```
int main()
{
   int a,b,c;
   b = 5;
   a = 3;
   a = 2* (b++);
   c = 2* (++b);
   return 0;
}
```

a) a = 10, c = 12
b) a = 10, c =5
c) a = 5, c = 10
d) a = 10, c = 14

13) The library function used to find the last occurrence of a character in a string is

A. strnstr(…)
B. laststr(…)
C. strrchr(…)
D. strstr(…)

14) When a break statement is encountered within a loop body,

A. The execution of the loop body is interrupted, and the program control transfers to the next statement following the loop.
B. All the remaining statements in the loop body are skipped and the loop continuation condition is evaluated next.
C. The program stops.
D. Nothing happens.

15) Given the following declarations

```
char c[50];
int i.
```

Select the expression that is equivalent to the following: c[i];

A. *(c+i)
B. &(c+i)
C. c+i
D. None of the above

16) In C, if you pass an array as an argument to a function, what actually gets passed?

A. Value of elements in array
B. First element of the array
C. First element address of the array
D. Address of the last element of array

17) Given the following declaration

**int i=1, j, *ip;**

Which of the flowing statements in **NOT** correct?

A. ip = &i;
B. j = *ip;
C. j = &ip;
D. (*ip)++;

18) What is the output of this C code?

```
void foo(int*);
int main(){
        int x = 20, *ptr = &x;
        foo(ptr++);
}
void foo(int *ptr){
        printf("%d\n", *ptr);
        }
```

A. 20
B. Some garbage value
C. Compile time error
D. Segmentation fault

19) The following function implements which searching or sorting algorithm

```
int algorithm (int a[], int n, int key)
{
    int i;
    for(i = 0; i < n; i++)
    {
        if (a[i] == key)
            return i;
    }
    return -1;
}
```

A. Bubble sort
B. Selection sort
C. Linear search
D. Binary search

20) What is the output of this C code?

```
int x = 0;
void main()
{
    int *ptr = &x;
    printf("%d\n", ptr);
    x++;
    printf("%d\n ", ptr);
}
```

A. Same address
B. Different address
C. Compile time error
D. Varies

**EXERCISE 3:** Write the output of the following C programs: ( /10 pts)

1)

```
#include <stdio.h>
int main()
{
    char s[100] = "Jeddah";
    char *p1 = &s[1];
    printf("The value of *p2 is %c\n", *++p1);
    char *p2 = &s[3];
    printf("The value of p2-p1 is %d \n", p2 - p1);
    return 0;
}
```

2)

```
#include <stdio.h>
void f(int i){
      i = i+2;
     }
void main()
      {
      int i=4;
      f(i);
      printf("The value is %d",i);
      }
```

3)

```
#include <stdio.h>
#include <stdlib.h>
void main()
{
  int i,*c = (int *)calloc(5,sizeof(int));
   for (i=0; i<5 ;i++)
      printf("%d, " , c[i]);
}
```

4)

```
#include <stdio.h>
#include <math.h>
int main ()
{
  float a = 11.2;
  int c = 5;
  float d = 4.0;
  printf("The result is %.1lf\n", fmod(a,c));
  printf("The result is %.1lf\n", sqrt(d));
 return(0);
}
```

5)

```
#include <stdio.h>
#include <math.h>
int main ()
{
  float val1=3.5, val2=4.2, val3=5.8;
  int val4 = 3;
  printf ("value1 = %.1lf\n", ceil(val1));
  printf ("value2 = %.1lf\n", ceil(val2));
  printf ("value3 = %.1lf\n", ceil(val3));
  printf ("value4 = %.1lf\n", ceil(val4));
  return(0);
}
```

**EXERCISE 4:** Answer the following questions **( /30pts)**

1- Declare a structure called **Employee** with the following elements: **name** as character array of size 15, ID as an int, **birthDate** as a structure whose elements are the integers day, month and year. (5pts)

2- Create a pointer to the structure Employee ( from part 1) called **EmpList** and dynamically allocate memory for 10 customers. (3pts)

3- Write the code to de-allocate memory used by the array **EmpList** (from part 2). (2 pts)

4. Write a function called printFile() that receives a file name as a string and print all the data in the file to the screen. ( /10pts)

5. Write the function **toUpper** that converts a lowercase character to an uppercase character. You may **NOT** use any other function inside this function.
Then in your main:
Ask the user to input a character to be converted to its uppercase.
Call the function tolower and pass the character entered by the user (10pts)