

First Semester 2017-18

Computer Programming CS F111

Comprehensive Examination

02-Dec-2017

Name: _____

80 Marks (40%)

2:00 – 5:00 PM

ID: _____

Closed Book

Part I

60 Min, 30M

General Instructions:

Part I is an in-built paper.

Please write your name and ID on the top right.

Assume all necessary header files are included in programs, even when not stated.

The ASCII value of 'A' is 65. For 'a' it is 97. For '0' it is 48.

I. Answer the following questions in the space provided.

- a. Assuming, a, b, c and min are four integer variables containing unequal values, write a single statement that stores the minimum of a, b, and c in min. [2]

- b. Identify if there is a bug in the code. If there is a bug, replace with the correct statement. [1.5]

```
#include <stdio.h>
int main(){
//Maximum possible value of int data type is 2147483647
int a = 1000000;
int c = 12345678;
long long int prod = a*c + 10;
printf("%lld\n",prod);
return 0;
}
```

- c. Why does the following code snippet produce the output "No" ? Briefly justify in a sentence. [1.5]

```
int main(){
float a= 5.5, b=5.11;
if((a/b)*5.11==25.5)
printf("Yes\n");
else
printf("No\n");
}
```

- d. Write suitable declarations for each of the following: [3]
1. ptr is pointer to an integer variable that is part of a struct NODE.

2. ptr is a two dimensional array (of size mxn) of pointers to character.

- e. What should X be in the following code snippet so that “Hello World” gets printed. [2]

```

if( X)
    printf("Hello");
else
    printf("World");

```

X =

II. Write the output of the following code snippets: [5]

<p>A.</p> <pre> main() { int x=0; for(; ;) { if (x ==10) break; x=x+1; printf("%d ",x); } } </pre>	<p>B.</p> <pre> main(){ for(int i=1;i<=5,printf("%d\t",i);printf("\td\t",i),i++); } </pre>
<p>E.</p> <pre> main(){ unsigned char i=0; int arr[256]={0}; for(i=0; i < 256; i++) if(i%50==0) printf("%d ", arr[i]); } </pre>	<p>D.</p> <pre> enum code { A, B, C}; main() { enum code c = 'Z' - 'Y'; int i = 2; switch{c} { case A: i += 20; case B: i -= 20; case C: i *= 20; default: i /= 20; } } </pre>
Answer A:	
Answer B:	
Answer C:	
Answer D:	
Answer E:	

III. Write the correct alternative(s) in the box provideds [8]

a. What would be the following snippet of code produce?

```
union details{
    int age;
    char mname;
} ;

union details d;
d.age = 8;
d.mname = 'R';
printf("%d", d.age);
```

- | | |
|---------------------------|------|
| A 114 | B 82 |
| C Cannot print as integer | D 81 |

b. Consider the following code snippet.

```
#include <stdio.h>
int main(){
char str1[ ] = "ComputerProgram";
char str2[ ] = "ComputerProgram";

(str1==str2)?(printf("Equal")):(printf("Unequal"));
return 0;
}
```

The output of the above piece of code is

- | | |
|--|---------------------|
| A Equal | B Unequal |
| C It will result in compile time error | D None of the above |

c. Which of the following is true?

- A malloc() and calloc() both allocate memory dynamically
- B There is no difference between malloc() and calloc(), except for number of parameters
- C Both A and B
- D Neither A nor B

d. Which of the following are correct ?

- A By pointer arithmetic it's possible to make a pointer point 503 bytes ahead from the current location it is pointing to.
- B Array of pointers to structures is not allowed in C.
- C A structure having two to five non-array members would occupy same number of bytes as that of a union with same members.
- D Unions cannot be nested within each other.

e. Choose the correct option based on the following piece of code.

```
typedef struct Exam
{
```

```

    char *name, *id;
    int q[10];
    float e1, e2;
    int labs[10];
    int online[2];

}CP;

int main(){
CP stu1, stu2;
stu1.name = (char*)malloc(sizeof(char)*4);
stu2.name = (char*)malloc(sizeof(char)*5);
strcpy(stu1.name, "TOM");
strcpy(stu2.name, "MARK");
printf("%d    %d", sizeof(stu1), sizeof(stu2));
return 0;
}

```

- A sizeof operator for stu2 will return a value of 1 greater than that returned for stu1.
- B sizeof function takes one argument
- C sizeof when applied to stu1.q results in 40 (assuming an int takes 4 bytes)
- D None of the above

f. What foo() does if n points to the first node of a singly linked list?

```

void foo(struct node * n){
    if(n == NULL)
        return;
    foo(n->next);
    printf("%d  ", n->data);
}

```

- A Prints the data of the list in reverse order
- B Prints all the elements of the list from head till the end.
- C Prints the alternate elements in the list
- D Removes the NULL element from the list

g. Consider following piece of code:

```

shape.rectangle.side1 = 10;

```

- A structure side1 is nested within structure rectangle
- B structure rectangle is nested within structure shape
- C structure shape is nested within structure rectangle
- D structure shape is nested within structure side1

h. The following line of code

```

#include<string.h>

```

gets replaced by the contents of the file string.h during

- A Linking
- B Execution
- C Preprocessing
- D Editing

IV. State True or False for each of the following. [4]

a) The following piece of code produces compile time error: _____

```
main() {  
    int * ptr = (int *) malloc(sizeof(int)*25);  
    int num = 25;  
    for(i = 25; i >= 0; i--) i[ptr] = 25-i;  
    free(ptr);  
    ptr = &num;  
}
```

b) Opening a file in write mode will create the file, if it does not exist. _____

c) In a conditional statement, 0 (zero) is considered as true and any non-zero value is false including negative values. _____

d) A do-while loop is guaranteed to run at least once. _____

e) The Linux command to change the path of a file is mv. _____

f) A function can have multiple return statements. _____

g) No two files in the Linux file system can have the same name. _____

h) There can be more than one variable with the same name in the same program.

V. A function computePerimeter() is written to compute the perimeter of a shape defined in struct shape. The function computePerimeter () is called inside main() as below:

```
void computePerimeter(struct shape *);  
main( ) {  
    struct shape *pt;  
    computePerimeter(pt);  
    printf("%f", pt->perimeter);  
}
```

Is there any major flaw in the logic used in the above code snippet? If so, how will you rectify it? [1]

VI. The program given below creates and uses what can be termed ragged arrays to store sixteen names input by the user. (The English word "ragged" means "having a rough or irregular surface or edge".) Diagrammatically explain the layout of the array variable list, and explain why it is justifiably called "ragged". [2]

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define MAX 16
int main()
{
    int num, i, len;
    char name[100], *list[MAX];
    for (i = 0; i < MAX; ++i)
    {
        scanf("%[^\n]", name);
        getchar(); /* to eat up the '\n' that remains */
        len = strlen(name);
        list[i] = malloc(len*sizeof(char));
        strcpy(list[i], name);
    } /* end of for loop */
    printf("\nReverse order name
    list:\n");
    for (i = MAX-1; i >= 0 ; --i)
        puts(list[i]);
    for (i = 0; i < MAX; ++i)
        free(list[i]);
}
```