

**University of Swaziland  
Faculty of Science and Engineering  
Department of Electrical and Electronic Engineering**

**Main Examination 2014**

---

**Title of Paper:** Programming Techniques I

**Course Number:** EE271

---

**Time Allowed:** 3 hrs

**Instructions:**

1. There are five (5) questions in this paper. Answer question 1 and any other three (3) questions.
2. Each question carries 25 marks.

**This paper should not be opened until permission has been given by the invigilator.**

**This paper contains thirteen (13) pages including this page.**

### Question 1

Choose the correct letter for the answer in each of the following options. Each correct answer or choice carries [1] mark.

a) Which of the following is a string?

- A. 'abcd'
- B. "abcd"
- C. abcd
- E. 'abcd'

b) The total memory required for an array is:

- A. sizeof (datatype) \* 2
- B. size of (datatype) \* size of used array elements
- C. sizeof (datatype) \* sizeof array
- D. size of (array) \* datatype

c) The keyword 'break' cannot be simply used within:

- A. do-while
- B. if-else
- C. for
- D. while

d) What is the correct order of precedence in C?

- A. addition, division, modulus
- B. addition, modulus, division
- C. multiplication, subtraction, modulus
- D. modulus, multiplication, subtraction

e) Compute the result of the following expression in 'C'.  $A=3*4/5+10/5+8-1+7/8$ . [1]

f) What is the minimum number of times that for loop is executed? [1]

g) Convert  $5271_8$  to Hexadecimal. Show all your working. [2]

h) What is the difference between passing arguments by reference and passing arguments by value? [2]

- i) Write a single statement that will print a random number from the set: [2]

4, 9, 16, 24, 36

- j) List the any two storage class specifiers in C. [2]

- k) What is the disadvantage of arrays in C? [1]

- l) True or False, the following are all valid variable names: `_under_bar_`, `m928134`, `t5`, `j7`, `her_sales`, `his_account_total`, `a`, `b`, `c`, `z`, `z2`. [1]

- m) When parentheses are nested, which set of parentheses is evaluated first in an arithmetic expression, innermost or outermost? [1]

- n) In a flow chart, what does the diamond symbol represent? [2]

- o) Assume `a = 3`, `b = 6`, `c = 8` and `d = 9`. What does the following statement print? [2]

```
printf("%d\t%d", a!=1 && d>=9, b==1 || c <=9);
```

- p) True or False. A 'C' program that totals the elements of a double-subscripted array must contain nested for statements. [1]

- q) What is the function of an interpreter program? [1]

Explain the following terms:

- r) Assembler. [1]

- s) Pseudocode [1]

## Question 2

- a) The following program compiles without errors, but does not produce the desired results when it is executed. It contains multiple logical errors. Fix the errors. **Note:** It is not necessary to rewrite the entire code – simple write the statements you have modified. [9]

```
#include <stdio.h>

int main(void)
{
    int i; /* loop counter */
    int type, score; /* inputs */
    int undergradTotal; /* sum of all scores of undergraduates */
    int gradTotal; /* sum of all scores of graduate students */
    int undergradCount; /* number of undergraduates */
    int gradCount; /* number of graduate students */

    for (i = 0; i < 100; ++i)
    {
        /* obtain inputs */

        printf("Enter student type (1 - undergrad, any other number -
            grad): ");
        scanf("%d", type);
        printf("Enter the students score: ");
        scanf("%d", score);

        /* update the appropriate total/counter combination */
        if (type = 1)
        {
            undergradTotal += score;
            ++undergradCount;
        }
        else
        {
            gradTotal += score;
            ++gradCount;
        }
    }

    /* produce output */
    if (undergradCount > 0)
        printf("Undergraduate Average: %.3f\n", undergradTotal /
            100);
    if (gradCount > 0)
        printf("Graduate Average: %.3f\n", gradTotal / 100);

    return 0;
}
```

b) What will be the output after the following program has been executed?

[10]

```
#include <stdio.h>

int main (void)
{
    int no = 1;
    int count;

    do
    {
        printf("%d\n",no++);
        count = no;

        while (count < 7)
        {
            printf("%d", ++count);
            if (count == 7)
                printf("\n");
        }
    }
    while (no < 6);

    return 0;
}
```

c) What will be the output after the following program has been executed?

[6]

```
#include <stdio.h>

void Rotate(int, int*, int);

int main(void)
{
    int x = 1;
    int y = 2;
    int z = 3;

    Rotate(x,&y, z);

    printf("%d %d %d\n", x, y, z);

    return 0;
}

void Rotate(int x, int *y, int z)
{
    int temp = x;

    x = *y;
    *y = z;
    z = temp;
}
```

### **Question 3**

A small airline has just purchased a computer for its new automated reservations system. The president has asked you to program the new system. You are to write a program to assign seats on each flight of the airline's only plane (capacity: 10 seats).

Your program should display the following menu of alternatives:

*Please type 1 for "first class"*

*Please type 2 for "economy"*

If the person types 1, then your program should assign a seat in the first class section (seats 1-5).  
If the person types 2, then your program should assign a seat in the economy section (seats 6-10).

Your program should then print a boarding pass indicating the person's seat number and whether it is in the first class or economy section of the plane.

Use a single-subscripted array to represent the seating chart of the plane. Initialize all the elements of the array to 0 to indicate that all seats are empty. As each seat is assigned, set the corresponding elements of the array to 1 to indicate that the seat is no longer available.

Your program should, of course, never assign a seat that has already been assigned. When the first class section is full, your program should ask the person if it is acceptable to be placed in the economy section (and vice versa). If yes, then make the appropriate seat assignment. If no, then print the message *"Next flight leaves in 3 hours."*

Your program will be graded according to the following criteria:

- i) Correctness – does the program produce the desired result. [18]
- ii) Clarity – proper indentation of program makes it easy to read. [2]
- iii) Sensible naming of variables – makes it easy to understand code when debugging. [3]
- iv) Proper use of comments – comments also make the program easy to understand. [2]

#### **Question 4**

- a) Convert the following code to use a “do-while” loop instead of for loop. [5]

```
#include <stdio.h>

int main (void)
{
    int i;

    for (i=25; i >=0; i--)
        printf("%c", 'z'-i);

    printf("\n");

    return 0;
}
```

- b) Write a complete C program that uses a **recursive function** named *Power*. The program should prompt the user to enter the base and the exponent values as integers, and then display the output computed by the function *Power*. The recursive function *Power(base, exponent)* when invoked returns

$$\text{base}^{\text{exponent}}$$

For example,  $\text{Power}(5, 4) = 5 * 5 * 5 * 5$ . Assume that exponent is an integer greater than or equal to 1. *Hint*: The recursion step would use the relationship

$$\text{base}^{\text{exponent}} = \text{base} * \text{base}^{\text{exponent} - 1}$$

and the terminating condition occurs when exponent is equal to 1 because

$$\text{base}^1 = \text{base}$$

**Note: You do not need to include comments in your program** [20]

### Question 5

Assume that single-precision floating point numbers are stored in 4 bytes, and that the starting address of the array is at location 1002500 in memory. Each part of this question should use the results of previous parts where appropriate.

- a) Define an array of type float called numbers with 10 elements, and initialise the elements to the values 0.0, 1.1, 2.2, 3.3, 4.4, 5.5, 6.6, 7.7, 8.8, 9.9. Assume the symbolic constant SIZE has been defined as 10. [2]
- b) Define a point, *nPtr*, that points to an object of type float. [1]
- c) Print the elements of array numbers. Print each number with 1 position of precision to the right of the decimal point. [4]
- d) Give two separate statements that assign the starting address of array numbers to the pointer variable *nPtr*. [3]
- e) Print the elements of array numbers using pointer/offset notation with the pointer *nPtr*. [5]
- f) This question is independent of the above. State the error in the each of the following program segments. Assume:

```
int *zPtr;
int *aPtr = NULL;
void *sPtr = NULL;
int number, i;
int z[5] = {1, 2, 3, 4, 5};
sPtr = z;
```

- i) `++zPtr;` [2]
- ii) `number = zPtr;` [2]
- iii) `number = *zPtr[2];` [2]
- iv) 

```
for (i = 0; i <= 5; ++i){
    printf("%d", zPtr[i]);
}
```

 [2]
- v) `number = *sPtr;` [2]