



# **CSC1003**

## **Programming Methodology**

---

### **Sample Quiz 2\_Guide**

#### **(1 hours 30 minutes)**

---

**Name:** \_\_\_\_\_  
**GUID/SIT number:** \_\_\_\_\_

#### **Instructions:**

- 1. Type your answer in docx file and name it as yourname\_SITnumber.docx**
- 2. Submit your docx file and three (3) .c code files electronically in your xSITE Dropbox under CSC1003 Programming Methodology.**

Answer all the questions (Total 100 marks)

Section A : Multiple Choice Questions. Answer all the questions (Total 40 marks)

### Question 1

Given a sample program below on pointer constant and pointer variable, determine the result of diff1, diff2 and the value of the pointer ptr\_y after the program has executed and stopped at the `return` statement. Assume memory address of y=0030

```
#include <stdio.h>

int main()
{
    static int
y[13]={1,2,4,8,16,32,64,128,256,512,1024,2048,4096};
    int i, diff1=128, diff2=128,a, *ptr_y;

    ptr_y=y;
    for (i=0;i<5;i++)
    {

        diff1 -=y[i];
        diff2 -=*(ptr_y+2*i)-y[i];
        ++ptr_y;
    }
    return 0;
}
```

- (A) diff1=97, diff2=-4522, ptr\_y=0046  
(B) diff1=97, diff2=-442, ptr\_y=0046  
(C) diff1=97, diff2=-4522, ptr\_y=0050  
(D) diff1=97, diff2=-442, ptr\_y=0050

(10 marks)

### Solution

	i=0	i=1	i=2	i=3	i=4
diff1	128-1=127	127-2=125	125-4=121	121-8=113	113-16=97
diff2	128-[* (30+0)-1] =128-(1-1)=128	128-[* (42)-2] =128-(8-2)=122	122-[* (54)-4] =122-(64-4)=62	62-[* (66)-8] =62-(512-8)=-442	-442-[* (78)-16] =-442-62=-504

					=-442-(4096-16)=-4522
ptr_y	34	38	42	46	50

## Question 2

Given a sample program on array of pointers, determine the total memory required to store the array of string pointers. Assume that 1 character occupies 1 byte of memory.

```
#include <stdio.h>
#include <string.h>

int main()
{
    static char
    *str_ptr[6]=
{"Singapore_", "Institute_", "of_", "Technology", "", "SIN"};
    int i;

    for(i=0; i<=5; i++)
    {
        printf("\n%s", str_ptr[i]);
    }
    return 0;
}
```

(A) 288 bits

(B) 336bits

(C) 36bits

(D) 6 bits

(10 marks)

## Solution

Number of bytes=11+10+4+11+2+4=42 bytes

No of bits =42x8=336 bits

### Question 3

Given a C program below called testing.c, the following command line is executed

.\\testing.exe 2 4 6 8 10 12 14 16 18 20

Determine the value of argc, argv[1] and the program output.

```
/* **** */
/* testing.c -command line arguments */
/* **** */

#include <stdio.h>
#include <stdlib.h>
int main(int argc, char **argv)
{
    printf("%d", atoi(argv[1])-atoi(argv[3]));
    return 0;
}
```

- (A) argc=10, argv[1]=2, -4
- (B) argc=11, argv[1]= .\\testing.exe , -4
- (C) argc=10, argv[1]=.\\testing.exe, -4
- (D) argc=11, argv[1]=2, -4

(10 marks)

### Solution

argv[1]=2  
argv[3]=6  
argv[1]- argv[3]=2-6= -4

#### Question 4

Given a C program on struct variables, determine the size of the struct variables location if outdoor=1 has been entered.

```
#include <stdio.h>
#define num 50
#define message01 "You are using SIT-WiFi"
#define message02 "You are using 5G data"
struct Point3D
{
    float x;float y;float z;
};
typedef struct Point3D WiFiPoint,Point5G;
typedef char *STRING;
struct SIT
{
    WiFiPoint SITWiFi[num];
    Point5G SIT5G[num];
    STRING Accesspoint;
};
struct SIT_location
{
    struct SIT SIT_NYP;
    struct SIT SIT_NP;
};
int main()
{
    int *ptr_choice,outdoor,k;
    char *display;
    struct SIT_location *ptr_location, location;
    ptr_choice=&outdoor;
    ptr_location=&location;
    printf("\nDetermine inside or outside building: ");
    scanf("%d",&outdoor);
    if (*ptr_choice==0)
        display=ptr_location->SIT_NP.Accesspoint=message01;
    if (*ptr_choice==1)
        display=ptr_location->SIT_NP.Accesspoint=message02;
    return 0;
}
```

- (A) 2422 bytes  
 (B) 2408 bytes  
 (C) 2416 bytes  
 (D) 2423 bytes

(10 marks)

### **Solution**

Number of bytes =  $2(50 \times 12 + 50 \times 12) + 22 = 2422$

*Section B : Written Response Questions. Answer all the questions (Total 60 marks)*

### **Question 5**

Write the pseudocode and C code using function and pointers to calculate the sample variance of the class performance for GCE O Level Mathematics examination. The sample variance  $s_x$  for set of student mathematics marks  $x_1 \cdots x_N$

$$s_x = \frac{1}{N-1} (\sum_{i=1}^N x_i^2 - N\bar{x}^2)$$

where  $x_i$  is the  $i^{th}$  student mathematics result and  $N$  is the number of students taking the examination. Write a main function that mainly

1. reads all number of students taking mathematics examination,  $N$  and the respective student result,  $x_i$
2. calls a statistical function to compute the sample variance  $s_x$
3. prints the sample variance  $s_x$

The statistical function should use pointer constant and variable to create sample variance  $s_x$  and return the result to the main function

(20 marks)

### **Sample Guide**

#### **Pseudocode**

```
BEGIN
  READ N
  FOR i = 0 to N - 1 do
    READ x[i]
  END FOR
  stats(x, N, &var)
  PRINT "sample size", N
  PRINT "Variance", Var
END
```

```

FUNCTION stats( $x, N, p\_var$ )
     $p\_var \leftarrow refTofloat \rightarrow \&var$ 

     $sumx \leftarrow x[0]$ 
     $sumx \leftarrow x[0] * x[0]$ 
    FOR  $i = 1$  to  $N - 1$  do
         $sumx \leftarrow sumx + x[i]$ 
         $sumxx \leftarrow sumxx + x[i] * x[i]$ 
    END FOR

     $p\_Mean \leftarrow sumx / N$ 
     $*p\_var \leftarrow (sumxx - N * (p\_Mean * p\_Mean)) / (N - 1)$ 
ENDFUNCTION

```

### C code

```

#include <stdio.h>
#include <math.h>

#define MAX_SIZE    500

void stats(float x[], int N, float *p_var);

int main()
{
    float x[MAX_SIZE];
    float var;
    int N, i;

    printf(" Enter number of student taking examination: ");
    scanf("%d", &N);
    for (i=0; i<N; ++i)
    {
        printf("Enter exam mark for student(%d):", i);
        scanf("%f", &x[i]);
    }
}

```

```

    }
    stats(x,N,&var);
    printf("\n Examination statistics\n");
    printf("\n No of student taking exam=%d\n",N);
    printf("\n Standard deviation of examination mark =%f\n",var);
}

void stats(float x[], int N,float *p_var)
{
    int i;
    float sumx, sumxx, p_mean;

    sumx=x[0];
    sumxx=x[0]*x[0];
    for(i=1;i<N;++i)
    {
        sumx +=x[i];
        sumxx +=x[i]*x[i];
    }
    p_mean=sumx/N;
    *p_var=(sumxx-N*(p_mean*p_mean))/(N-1);
}

```

## Question 6

Various matrix operation is essential tools for real time applications. For example, assume two matrices **A** and **B** such as followed,

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix}, \mathbf{B} = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \end{bmatrix}$$

we can obtain another third matrix **C** with the following matrix operation such as addition and subtraction. For addition operation

$$\mathbf{C} = \mathbf{A} + \mathbf{B}$$

The elements of matrix **C** are obtained as follows:  $c_{ij} = a_{ij} + b_{ij}$  for all  $i, j$

$$\mathbf{C} = \begin{bmatrix} a_{11} + b_{11} & a_{12} + b_{12} & a_{13} + b_{13} & a_{14} + b_{14} \\ a_{21} + b_{21} & a_{22} + b_{22} & a_{23} + b_{23} & a_{24} + b_{24} \\ a_{31} + b_{31} & a_{32} + b_{32} & a_{33} + b_{33} & a_{34} + b_{34} \end{bmatrix}$$

Write a C program where there is

1. a main function that



- a. call the `matrix_read` function to read in two matrix two matrices **A** and **B**
  - b. call the `matrix_ops` function to perform addition of two matrices A and B
  - c. print out the matrix operation result in matrix **C**
2. a `matrix_read` function that uses pointer constant and pointer variable to create the matrix with user input on the dimension of matrix in the function and return the matrix to the main function
3. a `matrix_ops` function that uses pointer constant to perform the matrix operation and return the computed matrix to the main function.

(20 marks)

### Sample Guide

#### C-Code

```
#include <stdio.h>

#define maxrow 10
#define maxcol 10

void matrix_read(float x[][maxcol],int *ptr_row, int *ptr_col);
void matrix_ops(float A[][maxcol], float B[][maxcol],float C[][maxcol],int
num_row,int num_col);
int main()
{
    int num_row, num_col,i=0,j=0;
    int choice=0;
    float A[maxrow][maxcol], B[maxrow][maxcol], C[maxrow][maxcol];

    matrix_read(A,&num_row,&num_col);
    matrix_read(B,&num_row,&num_col);
    matrix_ops(A,B,C,num_row,num_col);
    printf("\n");
    for (i=0;i<num_row;++i)
    {
        for (j=0;j<num_col;++j)
        {
            printf("%f ",C[i][j]);
        }
        printf("\n");
    }
}

void matrix_read(float x[][maxcol], int *ptr_row, int *ptr_col)
{
    int i,j;
    printf("\n number of rows in matrix: ");
    scanf("%d",ptr_row);
    printf("\n number of column in matrix: ");
    scanf("%d",ptr_col);
```

```

printf("\n Enter elements at a time");
for (i=0;i<*ptr_row;++i)
{
    for (j=0;j<*ptr_col;++j)
    {
        printf("\n Element %d %d = ",i+1,j+1);
        scanf("%f",&x[i][j]);
    }
}
}
void matrix_ops(float A[][maxcol],float B[][maxcol],float C[][maxcol],int
num_row,int num_col)
{
    int i,j;

    for (i=0;i<num_row;++i)
    {
        for (j=0;j<num_col;++j)
        {
            C[i][j]=A[i][j]+B[i][j];
        }
    }
}

```

### Question 7

Create a simple C program to key in the marks for the various assessment component of the Programming Methodology for the student class size,  $N$ . Your program should use structures to contain

1. Student name
2. Student GUID number
3. Assessment component namely
  - a. Quiz 1 score,
  - b. Quiz 2 score
  - c. Assessment Exercise (AE) Score and
  - d. Class Participation (CP) Score
  - e. Examination Score
4. Overall score

The program should read in the student size,  $N$  and key in all the student details, namely name, GUID and respective Assessment component score. The program is expected to calculate the overall mark obtained for the student with weightage of Quiz 1, Quiz 2, AE, CP and examination as 20%, 20%, 35%, 5% and 30% respectively.

(20 marks)

## Sample Guide

### C-Code

```
#include <stdio.h>

#define Quiz1_weight    0.15
#define Quiz2_weight    0.15
#define AE_weight       0.35
#define CP_weight       0.05
#define exam_weight     0.3
#define max_student     150

/*structure definition*/
struct grade_record
{
    char name[51];        /* student name*/
    char ID_num[51];      /* ID number*/
    float Quiz1;
    float Quiz2;
    float AE;
    float CP;
    float exam;
    float total;          /*total score*/
};

int main ()
{
    struct grade_record student[max_student];          /*array of structure*/
    float AE1_sum,AE2_sum,CP_sum,exam_sum,total_sum;    /*sum*/
    float AE1_avg,AE2_avg,CP_avg,exam_avg,total_avg;    /*average*/
    int i,num;

    /*initialize sum to zero*/
    AE1_sum=0;
    AE2_sum=0.0;
    CP_sum=0.0;
    exam_sum=0.0;
    total_sum=0.0;
    printf("\n Student Grades");
    printf("\n No. of students in class : ");
    scanf("%d",&num);
    /*get data for each student and compute total score and sum*/
    for (i=0;i<num;++i)
    {
        printf("\n \n Enter the student name: ");
        scanf("%s",student[i].name);
        printf("\n \n Enter the student ID: ");
        scanf("%s",student[i].ID_num);
        printf("\n \n Enter the Quiz1 marks: ");
```

```

scanf("%f",&student[i].Quiz1);
printf("\n \n Enter the Quiz2 marks: ");
scanf("%f",&student[i].Quiz2);
printf("\n \n Enter the AE marks: ");
scanf("%f",&student[i].AE);
printf("\n \n Enter the CP marks: ");
scanf("%f",&student[i].CP);
printf("\n \n Enter the exam marks: ");
scanf("%f",&student[i].exam);
/* compute total for this student*/

student[i].total=(student[i].Quiz1)*Quiz1_weight+(student[i].Quiz2)*Quiz2_weight+(s
tudent[i].AE)*AE_weight+(student[i].CP)*CP_weight+(student[i].exam)*exam_weight;
}
/*print heading and table*/

printf("\n \n -----");
printf("-----");
printf("\n NAME          ID          Quiz1          Quiz2          AE
CP          Exam          Total");
printf("\n -----");
printf("-----");

for(i=0;i<num;++i)
{
    /*print data for this student*/
    printf("\n %s\t %s\t",student[i].name,student[i].ID_num);

printf("%.2f\t\t %.2f\t\t %.2f\t\t %.2f\t\t %.2f\t\t",student[i].Qui
z1,student[i].Quiz2,student[i].AE, student[i].CP,student[i].exam,student[i].total);
}
printf("\n -----");
printf("-----");
}

```