NATIONAL UNIVERSITY OF SINGAPORE

SCHOOL OF COMPUTING

MID-SEMESTER TEST
AY2011/2012 Semester 1

CS1010 PROGRAMMING METHODOLOGY

8 October 2011 Time Allowed: **1 hour 30 minutes**

INSTRUCTIONS

1. This question paper contains **SIXTEEN (16)** questions and comprises **SEVEN (7)** printed pages.
2. An **ANSWER SHEET** is provided for you to write the answers. It comprises **TWO (2)** printed pages.
3. Answer **ALL** questions within the space provided on the **Answer Sheet**.
4. Maximum score is **30 marks**.
5. This is an **OPEN BOOK** test.
6. Write legibly with a pen or pencil.
7. Calculators, electronic dictionaries, laptops, PDAs and other computing devices are not allowed.
8. Submit only the **Answer Sheet** at the end of the test. You may keep the question paper.
9. Write your **MATRICULATION NUMBER** on page 1 of the **Answer Sheet** using a **PEN**.

——— END OF INSTRUCTIONS ———

SECTION A (10 Multiple Choice Questions : 10 Marks)

Each question has only one correct answer. Write your answers in the boxes provided on the **Answer Sheet**. 1 mark for each correct answer and no penalty for wrong answer.

1. Which of the following is/are not valid C identifiers?

   i) While
   ii) $&cts
   iii) ____

   A. (ii) only
   B. (iii) only
   C. (i) and (ii) only
   D. (ii) and (iii) only
   E. All (i), (ii) and (iii)

2. Suppose `x`, `y` and `z` are declared as integer variables with values 8, 6, and 2 respectively, what are the values of `x`, `y and z` after the following C statement is executed?

```
x /= y %= z ;
```

   A. It will give compile-time error.
   B. It will give run-time error.
   C. x is 2, y is 0, z is 2
   D. x is 0, y is 4, z is 2
   E. x is 5, y is 3, z is 2

3. In logic, there is a Boolean operator called the exclusive OR (XOR) which gives the result true if and only if one of the operands is true (see truth table below).

A	B	A XOR B
false	false	false
false	true	true
true	false	true
true	true	false

   Which expression is equivalent to the XOR operator?

   A. `(!A || !B)`
   B. `(!(A && B))`
   C. `((A && !B) && (!A && B))`
   D. `((A && B) || (!A && !B))`
   E. `((A || B) && (!A || !B))`

4. Based on the code below, which of the following statements can be used to replace the code in the box, to produce the same result?

```
sum = 0;
for (i = 0; i < n; i++)
    for (j = i; j < n; j++)
        sum++;
printf("%d\n", sum);
```

A. `sum += i;`
B. `sum += n - i;`
C. `sum += n;`
D. `sum *= n;`
E. `sum += n*(n-1)/2;`

5. What are the values of `sum` and `count` after the following loop terminates?

```
int sum = 0, count = 0;

do {
   count++;
   sum += count;
   if (sum > 6)
      break;
}
while (count < 6);
```

A. `sum` is 6, `count` is 3
B. `sum` is 10, `count` is 4
C. `sum` is 15, `count` is 5
D. `sum` is 21, `count` is 6
E. they are undefined

6. What is the output of the following program fragment?

```
int i=-3, j=2, k=0, m;
m = (++i && ++j) || ++k;
printf("%d, %d, %d, %d\n", i, j, k, m);
```

A. it will give compile-time error
B. `-3, 2, 0, 1`
C. `-2, 3, 0, 1`
D. `-2, 3, 1, 1`
E. `-2, 3, 1, 0`

7. Assuming i is declared as an integer, what is the output when you execute the following code (note: the ASCII code for 'A' is 65)?

```
i = (int)'A';
switch (i) {
   case 'A':
        printf("%d\n", i);
   case 'B':
        printf("%d\n", i);
   case 'C':
        printf("%d\n", i);
}
```

A. Compile time error
B. Only 65 will be printed
C. Only 'A' will be printed
D. 65 will be printed three times on separate lines
E. 'A' will be printed three times on separate lines

8. Which of the following statements about functions in C is/are true?

i) The default return type for functions is int

ii) Each executable C program must have exactly one main function

iii) If a function does not return any value, its return type is void

iv) You can have more than one return statement in a function definition

A. None of the above
B. (ii) only
C. (i) and (iii) only
D. (ii) and (iii) only
E. All of the above are true

9. What is printed out by the following code fragment?

```
int arr[] = {1, 2, 3, 4, 5};
printf("%d\n", arr[arr[arr[arr[1]]]]);
```

A. It will give compile-time error
B. 1
C. 3
D. 5
E. out of range error

10. What is printed out by the following code fragment?

```
int i = -40, j = 0;

if (-50 < i < -30)
    j = 1;
printf("%d\n", j);
```

A. It will give compile-time error
B. 0
C. 1
D. NaN
E. An undefined value

Section B: Short Structured Questions (10 Marks)

Write your answer in the space provided on the **Answer Sheet**.

11. For each of the following statements that appears in the body of the main function, indicate whether it will give rise to a warning message or a syntax error. For those that do, suggest a suitable correction by writing out the correct statement. [3 marks]

i) `int arr[5] = {0.0, 1.1, 2.2, 3.3, 4.4, 5.5};`

ii) `int p*;`

iii) `char c = "A";`

12. What is the output of the following code? [2 marks]

```
#include <stdio.h>

int main(void)
{
    int i, n = 1024;
    int count = 0;

    for (i=0; i < n; i = i*2)
       count++;
    printf("%d %d\n", n, count);
    return 0;
}
```

13. What is the output of the following code? [2 marks]

```
#include <stdio.h>

int main(void)
{
     int i, list[5]={0};

     for (i=0; i<5; i++)
          list[list[i]]++;

     for (i=0; i<5; i++)
         printf("%d ", list[i]);

     return 0;
}
```

14. What is the output of the following program? [3 marks]

```
#include <stdio.h>

void figure_me_out(int *, int, int *);

int main(void)
{
     int a=10, b=20, c=30;
     figure_me_out(&a, b, &c);
     printf("%d %d %d\n", a, b, c);
     return 0;
}

void figure_me_out(int *i, int j, int *k)
{
     int a, b=j;
     for (a=*i; a<50; a+=b);
     (*k)++;

}
```

Section C: Short Programming Question (10 Marks)

Write your answer in the space provided on the **Answer Sheet**.

15. The Babylonian algorithm to compute the square root of a number ***n*** is as follows:

    1. Make a guess at the answer (you can pick ***n*/2** as your initial guess).
    2. Compute ***r = n/guess***
    3. Set ***guess = (guess+r)/2***
    4. Go back to step 2 for as many iterations as necessary. The more times steps 2 and 3 are repeated, the closer guess will become to the square root of ***n***.

    Write a function that accepts a positive integer for ***n***, iterates through the Babylonian algorithm until guess is within 1% of the previous guess and returns the answer as a value of type double. [4 marks]

16. Assuming `arr` is an array of integers and `n` is size of the array, describe the purpose of the following function? Keep your answer concise. [2 marks]

```
int do_something(int arr[], int n)
{
   int i, j;
     for (i = 0; i < n; i++)
       for (j = 0; j < n; j++)
          if (i != j && arr[i] == arr[j])
                    return 1;
     return 0;
}
```

    Write an improved version of the function to make it more efficient. Call this function `do_something_better`. [4 marks]

——— END OF PAPER ———