UNIVERSITI TEKNOLOGI MALAYSIA

TEST 1

SEMESTER I 2014/2015

SUBJECT CODE	**: SCJ1013 / SCSJ1013**
SUBJECT NAME	**: PROGRAMMING TECHNIQUE I**
YEAR/COURSE	**: 1 (SCSJ / SCSV / SCSB / SCSR)**
TIME	**: 2 HOURS AND 30 MINUTES**
DATE	**: 7 NOVEMBER 2014**
VENUE	**: N28 BK1-BK6**

INSTRUCTIONS TO THE STUDENTS:

This test book consists of 3 parts:

PART A : Basics of C++	**[29 marks]**
PART B : Selection Structure	**[25 marks]**
PART C : Loop Structure	**[31 marks]**
TOTAL	**[85 marks]**

ANSWER ALL QUESTIONS IN THIS BOOKLET IN THE SPACES PROVIDED.

Name	
I/C No.	
Year/Course	
Section	
Lecturer's Name	

This question booklet consists of 15 pages including the cover page.

PART A : Basics of C++ **[29 marks]**

1. (a) Label the order of execution for each operator in the following expression. The operator that is evaluated first should be labeled as 1, the second operator to be evaluated as 2 and so on. (1.5 marks)

z	=	(	10	+	0.4	)	/	2	+	25	/	4	-	x	++

(b) Given : **a = 10** , **b = 20** , **c = 15** , **d = 8.** Solve the following arithmetic expression by showing the order of execution of each operator. Assume all variables are of type `int`. (3 marks)

```
x = a * b / (-c * 31 % 13) * d;
```

(c) Given the declaration :

```
double first = 4.7;
int    second = 27;
```

What are the values of the variables **first** and **second,** and the respective expression after the execution of each statement below? Assume that each statement is executed independently. Write your answers with an accurate decimal point whenever required.

(6 marks)

No.	Statement	Value after execution		
		first	**second**	Expression
1.	`first=first + second;`			
2.	`second=first + second;`			
3.	`static_cast<int> (first + second);`			
4.	`second= second % static_cast<int>first;`			
5.	`first=second % static_cast<int> (first);`			
6.	`first=second + static_cast<int>(first);`			

2. What is the output of **`Program A.1`** given below? (5 marks)

```
 1 // Program A.1
 2 #include <iostream>
 3 using namespace std;
 4
 5 int main()
 6 { int a, c, d;
 7   double b;
 8   int count=4, x=15, y=18;
 9   double w= -17.3;
10
11   a= --count;
12   b= w++;
13   c= (x+count++ -y);
14   d= a+b+c;
15
16  cout <<"A=" << a << "  B="<<b
17       <<"  C="<<c  <<"  D="  <<d
18       <<"  COUNT="<<count;
19
20  return 0;
21 }
```

Output

A=_____ B=_____ C=_____ D=_____ COUNT=_____

3. Trace **`Program A.2`** by writing the output in the boxes provided in **Table 1** below. Write one digit, symbol or empty space per cell. (5.5 marks)

```
 1  // Program A.2
 2  #include <iostream>
 3  #include <iomanip>
 4  using namespace std;
 5
 6  int main()
 7  {  float x= 23.12767, y = 1345.4567, z = 866.887234, q = 2.34;
 8
 9    cout << setprecision(4) << x << endl;
10    cout << setprecision(5) << x << endl;
11    cout << setprecision(2) << x <<  endl;
12
13    cout << fixed<<setprecision(2) ;
14    cout << setw(7)<<x<<endl;
15    cout << setw(7)<<y<<endl;
16    cout << setw(7)<<z<<endl;
17
18    cout << fixed<<showpoint;
19    cout<<setprecision(1) ;
20    cout << x<< endl<<y<<endl<< z<<endl;
21
22    cout<<setprecision(6) ;
23    cout<<q<<endl;
24    cout<<showpoint<<q<<endl;
25
26    return 0;     }
```

Table 1

4. **Program A.3** below is meant to ask the user to enter two points; (x1, y1) and (x2, y2), respectively, and display their distance. The distance between the two points is expressed by the following formula:

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Complete **Program A.3** based on the instructions or comments written in (a) to (e).

(8 marks)

```
 1 // Program A.3
 2 #include <iostream>
 3 #include <iomanip>
 4 #include <cstdlib>
 5 #include <cmath>
 6 using namespace std;
 7
 8 int main()
 9 {
10 // (a) Declare  x1, y1 ,x2 and y2 as double variables  (2m)
11 _____________________________;
12
13 // (b) Declare distance as double variable           (0.5m)
14 _____________________________;
15
16 // Display a corresponding prompt to the user
17 cout << "Enter four double for value of x1,x2,y1,and y2";
18
19 // (c) Read all the required data from the user and
20 //       store them into variables x1, y1, x2, and y2,
21 //       respectively.                                (2m)
22 _____________________________;
23
24 // (d) Calculate the distance between the two points.  (3m)
25 ____________________________________________________;
26
27 // (e) Display the distance.                          (0.5m)
28 ____________________________________________________;
29
30   return 0;
31 }
```

PART B : Selection Structure [25 marks]

1. Assume a program has the following declarations.

```
char c = 'a', d = 'z';
bool flag = false;
int x = 5, y = 7, z = 10;
```

Based on these declarations, evaluate the following expressions:- (4 marks)

a) `(!flag) && (x < y)`

b) `(c < d) || (z > y)`

c) `(x == 5) && (z > 10)`

d) `x != 5`

2. A program segment is given as below :

```
int day;    // The number of days late

float fine; // The amount of fine charged

cin >> day;

if (day > 3)
   fine = 6.00;
else if (day >= 7)
   fine = 10.00;
else if (day >= 10)
   fine = 20.00;
cout << "Your penalty charge is RM" << fine;
```

a) What is the output of the program segment, if the value entered by the user for `day` is **`10`**? (1 mark)

b) The given program segment is actually meant for calculating the fine charged for the late return of library book(s). The fine rate will be determined according to **Table 2**. Rewrite the above program segment so that it will accomplish the user requirement as stated in the table. (3 marks)

Table 2

Number of days late	Fine Charged
Less than 3 days	No charge
Between 3 days to one week	RM6.00
More than 1 week and up to 10 days	RM10.00
More than 10 days	RM20.00

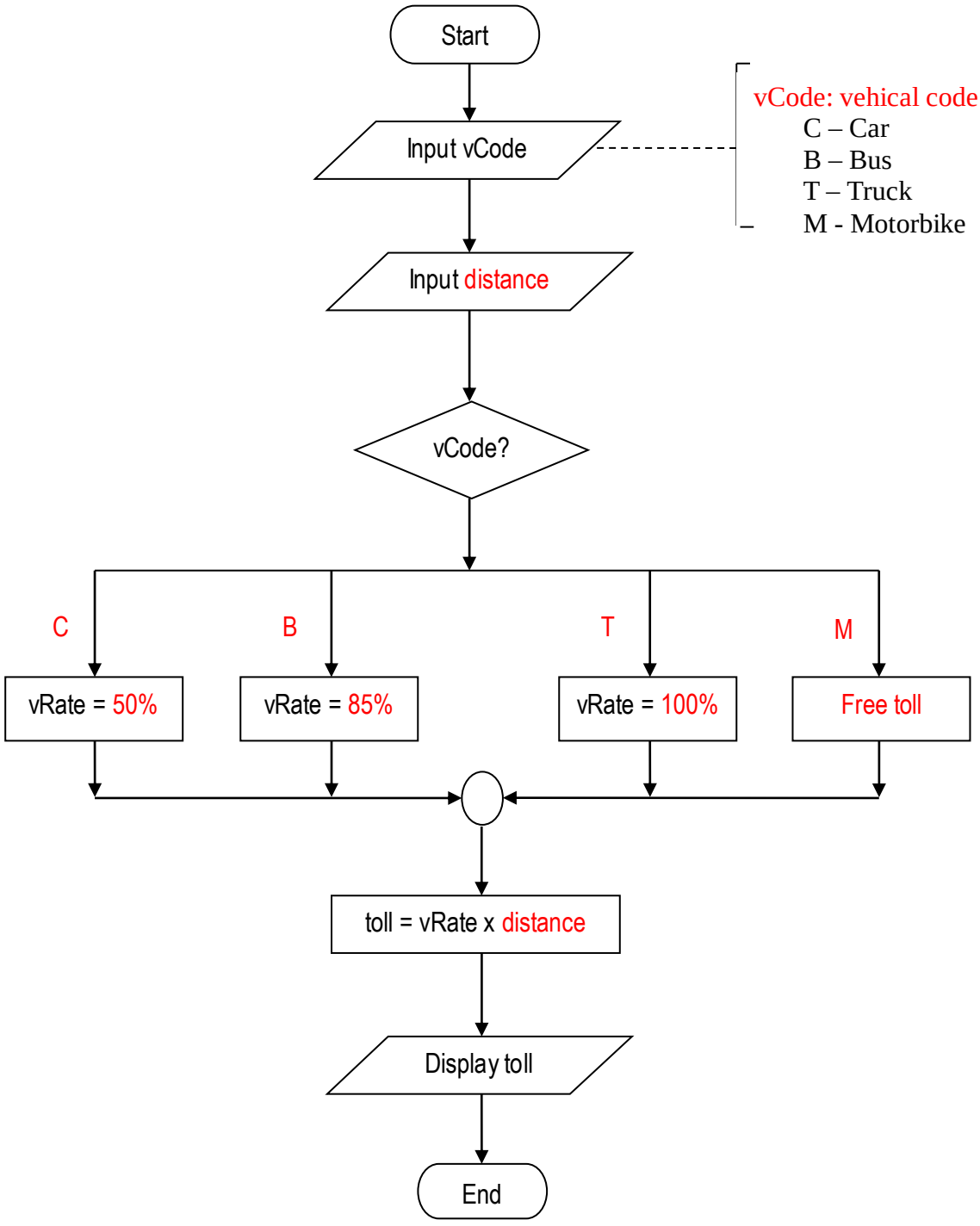
3. Trace **`Program B.1`** as given below and state the output at each iteration of the loop. Finally show the complete output that will be displayed by the program. (5 marks)

```
 1  // Program B.1
 2  #include <iostream>
 3  using namespace std;
 4
 5  int main()
 6  {
 7      for (int i = 0; i < 4; i++)
 8      {
 9          switch (i)
10          {
11              case 0  : cout << "A";
12              case 1  : cout << "B"; continue;
13              case 2  : cout << "C"; break;
14              default : cout << "D"; break;
15           }
16           cout << "E" << endl;
17       }
18       return 0;
19  }
```

i	Output

The final output for this program is :

4. Translate **Flowchart 1** into C++ code using switch statement(s). (7 marks)



Flowchart 1

```
// Program B.2
#include <iostream>
#include <iomanip>
using namespace std;

int main()
{   char vCode;  // Vehicle code: C –Car, B –Bus,
                 //                T –Truck, M -Motorbike
    double vRate; // Toll rate for the vehicle
    double toll, distance;









     return 0;
}
```

5. Complete **`Program B.3`** that will fulfill the requirements for conversion from Malaysian Ringgit to any of the currencies listed in **Table 3**. Your program should allow the user to choose an option from the list of options as stated in **Table 3**. To accomplish this, the respective conversion factors of the Malaysian ringgit to a particular currency is supplied as in **Table 4**. (5 marks)

Table 3

1. US Dollar
2. British Pound
3. Singapore Dollar
4. Indonesian Rupiah
5. Saudi Riyal

Table 4

The conversion factors are:
1 Malaysian Ringgit (MYR)
= 0.306 US Dollar (USD)
= 0.192 British Pound (GBP)
= 0.39 Singapore Dollar (SGD)
= 3737 Indonesian Rupiah (IDR)
= 1.15 Saudi Riyal (SAR)

Fill in the blanks in order to complete the program that will perform the above requirements.

```
// Program B.3
#include <iostream>
#include <iomanip>
using namespace std;

int main()
{
    int currency;
    double MYR, USD, GBP, SGD, AUD, IDR, SAR, rate;

    cout << "Enter a choice for exchange ";
    cin >> currency;

    cout << fixed << showpoint << setprecision(2);
    cout << "How many ringgit do you want to exchange? RM";
    cin >> MYR;
```

```
    if (_____________________________)
    {  _____________________________;
       _____________________________;
       cout << "You will be getting USD" << _____ << endl;  }
    else if (_____________________________)
    {  _____________________________;
       _____________________________;
       cout << "You will be getting GBP" << _____ << endl;  }
    else if (_____________________________)
    {  _____________________________;
       _____________________________;
       cout << "You will be getting SGD" << _____ << endl;  }
    else if (_____________________________)
    {  _____________________________;
       _____________________________;
       cout << "You will be getting IDR" << _____ << endl;  }
    else if (_____________________________)
    {  _____________________________;
       _____________________________;
       cout << "You will be getting SAR" << _____ << endl;  }

    return 0;
}
```

PART C : Loop Structure [31 marks]

1. Determine the output of each code segment below. (6 marks)

(a).

```
int x=10;
do {
   x++;
   if (x % 2 == 1  ) continue;
   cout << x << ", ";
} while (x<=21);
```

Output

(b)

```
int y=2;
do {
   if (y > 500 ) break;
   cout << y << ", ";
   y *= 3;

} while (y>1);
```

2. Using the **`for`** statement, write a code segment that prints the factorial of an integer number. The number is entered by the user. The factorial of an integer number , *n*, is expressed by

$$n! = n \times (n-1) \times (n-2) \times (n-3) \ldots \times 1$$

For example the factorial of number 5 is

$$5! = 5 \times 4 \times 3 \times 2 \times 1 = 120.$$

(5 marks)

Note: Write your answer in the box on the next page.

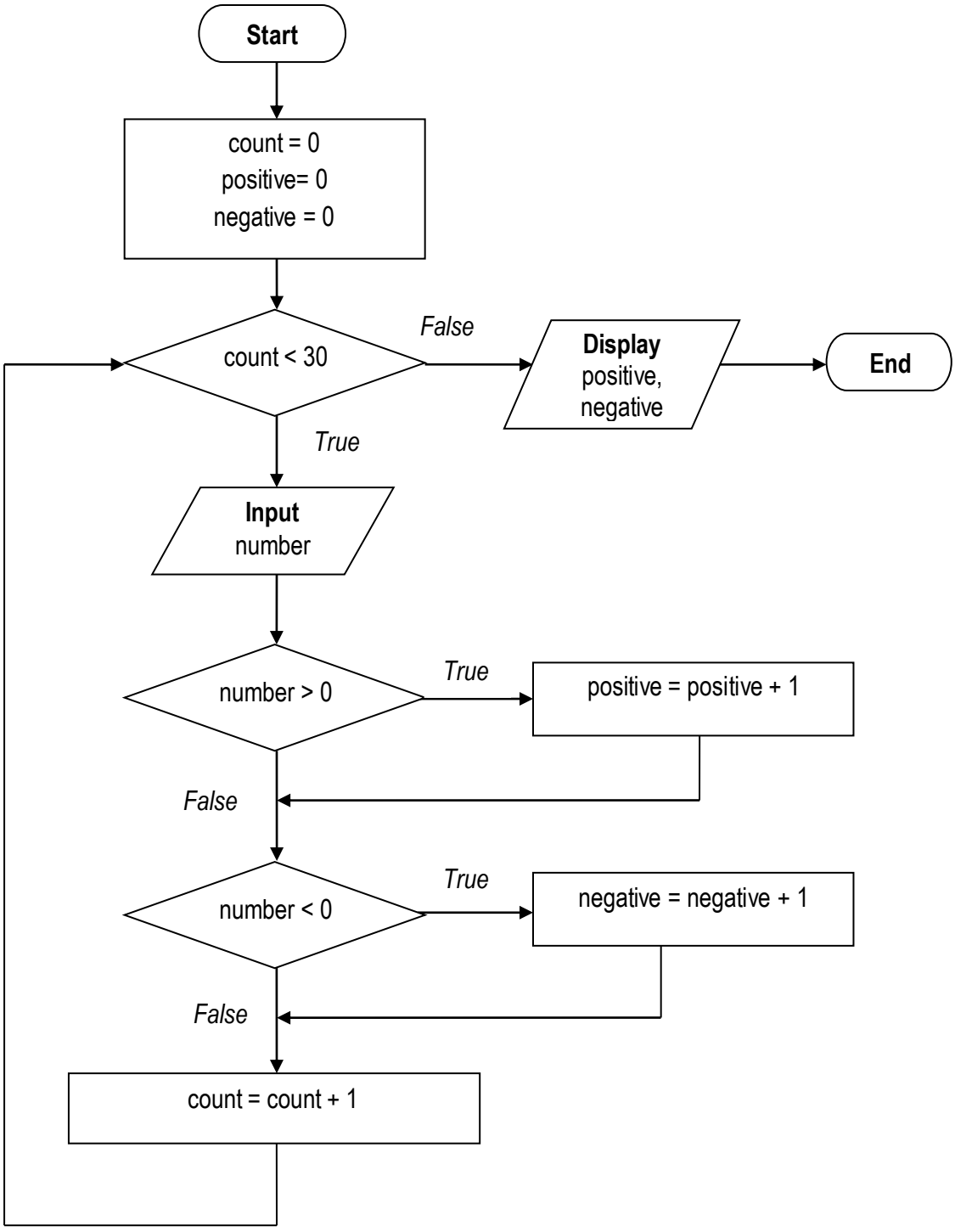
3. Using the **`while`** statement, write a code segment to print the *n*-th number in the Fibonacci series. The first number is when *n=1*, second number is when *n=2*, and so forth. The value of *n* should be obtained from the user. The Fibonacci series of numbers are

   0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

   Starting from the third number, a number in the series is determined by adding up the two numbers before it. For example the fifth number is determined by adding the third and fourth number, i.e., 3=1+ 2. ***Note:*** For simplicity, assume that the code only considers for $n \geq 3$

   (6 marks)

4. **Flowchart 2** presents an algorithm of a program that counts the number of positive and negative numbers entered by the user. The user will enter 30 numbers. Translate the flowchart into its corresponding C++ program using an appropriate loop statement. Write your answer by completing **Program C.1** follows. (5 marks)



Flowchart 2

```
// Program C.1
#include <iostream>
using namespace std;

int main()
{
     int count = 0;         // The total count of numbers entered
     int positive = 0;      // The count of positive numbers
     int negative = 0;      // The count of negative numbers
     int number;            // The number entered









      return 0;
}
```

5. **Program C.2** is incomplete. It is meant to be used by a cashier in a supermarket. It will ask the cashier to enter the prices and the number of items purchased by a customer. When there are no more items to key in, the cashier will enter a zero value for the item's price. Finally, the program will print a customer's receipt on the screen telling the total amount of purchase and the price of the most expensive item purchased by the customer. **Figure C.1** shows what the program should look like when it runs. Note that the **bold texts** are user inputs.

```
Enter the item's price: 20.80
Enter the number of item: 5

Enter the item's price: 18.90
Enter the number of item: 1

Enter the item's price: 26.00
Enter the number of item: 2

Enter the item's price: 0


Total Payment: RM 174.90
The price of the most expensive item: RM 26.00
```

Figure C.1

Based on all the information given, complete the program. (9 marks)

```
// Program C.2
#include<iostream>
#include<iomanip>
using namespace std;

int main()
{   double itemPrice;     // The price for each item purchased
    int itemCount;        // The count for each item purchased
    double totalPayment;  // The total amount of payment

    double highestPrice;  // The price of the most expensive item
                          //    purchased by the customer.

    cout << showpoint << fixed << setprecision(2);
```

```
    return 0;
}
```