REVISION TEST 1 (SECJ1023)

PROGRAMMING TECHNIQUE II

SECTION 03 & 04, SEM 2, 2020/2021

Question 1 **[22 marks]**

a) Write a class declaration named **`Inventory`** that has **`itemNumber`** (which contains the id number of a product) and **`numOfItem`** (which contains the quantity on hand of the corresponding product) as private data members. The default constructor to the **`Inventory`** class initializes the **`itemNumber`** and **`numOfItem`** member to **`0`**. Add an overloaded constructor to the **`Inventory`** class. This constructor should accept two arguments and design its value to the **`itemNumber`** and **`numOfItem`** member variables. Write three public inline member functions named **`getId`** (this member function puts the value of **`id`** into the private data member **`itemNumber`**), **`getAmount`** (this member function puts the value of **`num`** into the private data member **`numOfItem`**), and **`display`** (this member function displays the value of the object to the screen). (14 marks)

b) Write a C++ statement that defines an array of 10 objects of the **`Inventory`** class in Question (a), name this object as **`products`**. (1 mark)

c) Write C++ statements that will read inventory numbers and number of items from the keyboard (inputs will be entered by the user) into the array of objects represented by the array you defined in Question (b). There should be calls to both **`getId`** and **`getAmount`** member functions. (5 marks)

d) Write C++ statements to print out the values (**`itemNumber`** and **`numOfItem`**) for each object in the array that you defined in Question (b). This should be done by calling the member function **`display`** within a loop. (2 marks)

```
a) class Inventory  //0.5M         (14 marks)
   {
     public:
       Inventory()
       { itemNumber = 0;
         numOfItem = 0; }

       Inventory(int a, int b)
       { itemNumber = a;
         numOfItem = b; }

       void getId(int id) { itemNumber = id; }    //2M
       void getAmount(int num) { numOfItem = num; } //2M

       void display()//1M
       {
             cout <<"Item number " << itemNumber << " has "
            << numOfItem << " items in stock " << endl; }

     private:
       int  itemNumber;
       int  numOfItem;
   };
```

```
b) Inventory products[10];  //1M   (1 mark)

c) for (int i = 0; i < 10; i++) //1M       (5 marks)
   {
          cin >> id >> num; //2M
          products[i].getId(id); //1M
          products[i].getAmount(num); //1M
   }

d) for (int i = 0; i < 10; i++) //1M
            products[i].display();//1M
```

Question 2 **[13 marks]**

Given Program 1 as follows; please answers Question 2 parts (a) to (g).

```
#include <iostream>
#include "Stock.h"
using namespace std;

class Book
{
   private:
      static int numBook = 0;
      static double pricePay = 0;
      double price;  //Book’s price

   public:
      Book(double);
      double getPrice()
      {  return price;  }
      int getNumBook()
      {  return numBook;  }
      double getPricePay()
      {  return pricePay;  }

      /*(b)Insert static inline member function named
        totPrice to calculate the total price of books. */

      //(f)Declare the addStock friend function here.
};

Book::Book(double b)
{
   numBook++;
   price = b;
}

//Insert new statement(s) if necessary.

int main()
{
   double price;
   cout << "Enter your book price: RM";
   cin >> price;
   //(c)Call the totPrice member function

   Book obj1(50.5);
   Book obj2(21.0);
   /*(d)Use the obj1 and obj2 object to call the totPrice
      member function  */

   cout << "Total book that you have purchased: ‟ <<
   obj1.get getNumBook() << endl;
   cout << ‟Total price that you need to pay: RM” <<
   obj2.getPricePay() << endl;

   //(g)Define obj3 and initialize with obj1’s values.
   Return 0;
}
```

Program 1

a) Identify the error(s) in the class book declaration and correct it, by modifying the statement(s) in the class book declaration and inserting new statement(s) as indicated in Program 1. Provide your answers by completing the following table: (4 marks)

Line	A Corrected/ New Statement

b) At line 21, write a **`static`** inline member function named **`totPrice`** to calculate the total price of books. (2 marks)

c) At line 40, write a statement to call the **`totPrice`** member function before any instances of the **`Book`** class is created. (1 mark)

d) At line 44, write statements to call the **`totPrice`** member function by using the objects of the **`Book`** class created in line 42 and 43. (2 marks)

e) Assuming the error(s) is fixed in Question (a), and the function and statements in Question (b) to (d) have successfully been written. Based on the input given, what is the output of the program? (2 marks)

Input: `Enter your book price: RM`**`50.00`**

f) At line 24, write a friend function declaration named **`addStock`** that belongs to the **`Stock`** class. Notice the function takes two arguments, a double and an integer values; and does not return any value. (1 mark)

g) At line 52, define an object named **`obj3`** and initialize it to the values stored in **`obj1`**. (1 mark)

a) (4 marks)

Line	A Corrected/ New Statement
8	static int numBook; //1M
9	static double pricePay; //1M
33	int Book::numBook = 0; //1M
34	double Book::pricePay = 0.0; //1M

b) `static void totPrice(double a)` //1M (2 marks)
`{ pricePay += a; }` //1M

c) `Book::totPrice(price);` //1M (1 mark)

d) `obj1.totPrice(obj1.getPrice());`//1M
`obj2.totPrice(obj2.getPrice());`//1M (2 marks)

e) `Total book that you have buy: 2` //1M
`Total price that you need to pay: RM121.5` //1M (2 marks)

f) `friend void Stock::addStock(double, int);` //1M (1 mark)

g) `Book obj3 = obj1;` //1M (1 mark)

Question 3 **[14 marks]**

Given Program 2 below. Please answer Question 3 parts (a) to (d).

```
1   #include <iostream>
2   using namespace std;
3
4   class Line
5   {
6     public:
7       Line(int p)
8       {
9         cout << "Normal constructor allocating ptr" << endl;
10        ptr = new int;
11        *ptr = p;
12      }
13
14      //(a) Define a copy constructor
15      //(b)Define a destructor
16      //(c)Define an overloaded operator for the operator +
17
18      int getPtr(){ return *ptr; }
19      void setPtr(int p){ *ptr = p; }
20
21    private:
22      int *ptr;
23  };
24
25  int main()
26  {  Line obj1(10);
27     Line obj2 = obj1;
28     obj2.setPtr(15);
29     cout << obj1.getPtr() << endl;
30     cout << obj2 + obj1 << endl;
31     obj1.~Line();
32
33     return 0;   }
34
```

Program 2

a) At line 14, define a copy constructor that copies the value of pointer **`ptr`** from other **`Line`** and display the message **`"Copy constructor allocating ptr"`**. (4 marks)

b) At line 16, define a destructor to release a memory and display the message **`"Freeing memory!"`**. (3 marks)

c) At line 18, define an overloaded operator for the operator + such that when two lines, *e.g.* **`y`** and **`z`**, are added, the result that will be returned is the sum of the pointer **`ptr`** of both lines. For example, if the values of pointer **`y.ptr`** and pointer **`z.ptr`** are 5 and 3, respectively, the result will be returned for the operation **`y`** + **`z`** is 8. (2 marks)

d) Assuming all the definitions in Question 3 (a) to (c) have successfully been written, what is the output of the program? (5 marks)

a) (4 marks)

```
Line(const Line &obj) { //1M
   ptr = new int; //1M
   *ptr = *obj.ptr; //1M
     cout << "Copy constructor allocating ptr" << endl; //1M
}
```

b) (3 marks)

```
~Line() { //1M
   delete ptr; //1M
   cout << "Freeing memory!" << endl; //1M
 }
```

c) (2 marks)

```
int operator+(const Line &right) //1M
{ return *ptr + *right.ptr; } //1M
```

d) (5 marks)

```
Normal constructor allocating ptr //1M
Copy constructor allocating ptr //1M
10 //1M
25 //1M
Freeing memory! //1M
```

Question 4 [12 marks]

Answer all questions in this section based on the declaration of **`time`** class in Figure 1.0.

```
1   class time
2   {     int hour, minute, second;
3     public:
4         time(int hr=00, int mn=00, int sc=00)
5         { hour = hr; minute = mn; second = sc; }
6         time()
7         { hour = 16; minute = 30; second = 30; }
8         ~time() { cout << "Destroying precious time";
9         void setHour(int hr){ hour = hr; }
10        int getHour(){ return hour; }
11        friend void increaseTime(time);
12  };
```

Figure 1.0 **`time`** class declaration

1. Based on each term given below, give the line/lines of source codes in Figure 1.0 that implement the concept. Give the purpose of each concept. [10 marks]

Term	**Line/lines of codes**	**Purpose**
a. Constructor overload.		
b. Destructor.		
c. Inline function.		
d. Mutator function		
e. **`friend`** function		

2. Explain the error caused by the constructor declarations in Figure 1.0. Describe how the error can be corrected. [2 marks]

(1)

Term	Line/lines of codes	Purpose
a. Constructor overload.	Line 4 and 6	Flexible in declaring and initializing instance variable in various ways
b. Destructor.	Line 8	To release object memory
c. Inline function.	Line 9,10	Execution time of a program will be faster, compiler can replace those function **definition/codes** wherever the functions are being called.
d. Mutator function	Line 9	The function change values for private attribute
e. **`friend`** function	Line 11	Enable the function to access the private member of certain class that declared it as friend

(2)

- call of overloaded 'time()' is ambiguous
- Remove constructor in line 6 or change the type of parameter that differ from parameter of the constructor in line 4.

Question 5 **[15 marks]**

Answer all questions in this section, based on the **`lift`** program in Program 1.0.

```
// Program 1.0
#include <iostream>
using namespace std;
class lift
{      int level;    // the level where the lift is currently at
       int maxLoad; // the maximum load allowed
       int load;    // the current load in the lift
public:
       lift (int l=1,int lod=500,int ml = 10000);
       // member function prototype declaration
       void resetLift(int x,int y) {maxLoad = x; level=y;}
       void enterLift(int);  //people or goods enter the lift
       void move(int); // the lift go to the specified level
       void display();
       ~lift(){cout << "End of lift Object \n";}
};  // end lift class declaration

lift::lift (int l,int lod,int ml)
{ level = l;
  load = lod;
  maxLoad = ml;
}

void lift::enterLift(int load2)
{ if ((load+load2)> maxLoad)
    cout << "OVERLOAD, Lift cannot move. \n";
  else
  {  load += load2;
     cout << "SUFFICIENT LOAD. Moving..\n";
  }
}

void lift::move(int a)
{ if (a < level)
    cout << "Goes down!\n";
  else if (a > level)
     cout << "Goes UP!\n";
    else
     cout << "Same level\n";
    level = a;
}

void lift::display()
{ cout << "Current lift level " << level << "\n";
  cout << "Current maximum Load " << maxLoad << "\n";
  cout << "Current Load " << load << "\n\n";
}

void changeSpec(lift x)
```

```
50  {    x.resetLift(20, 20000);
51       x.display();
52  }
53
54  main()
55  { lift l1(5,800);
56    l1.display();
57    l1.enterLift(1000);
58    l1.move(10);
59    l1.display();
60    changeSpec(l1);
61    l1.display();
62    return 0;
63  } // end main()
```

a) Give the output produced by the following lines. [12 marks]

Line	Answer
56	
57-58	
59	
60	
61	
62	

b) Rewrite **changeSpec(lift x)** so that the function able to implement pass by reference using pointer concept (line 49-51). [2 marks]

c) Rewrite line 60 to implement pass by reference in (b). [1 mark]

(a)

Line	Answer
56	Current lift level 5 Current maximum Load 10000 Current Load 800 1.5 marks
57-58	SUFFICIENT LOAD. Moving.. Goes UP! 2 marks
59	Current lift level 10 Current maximum Load 10000 Current Load 1800 1.5 marks
60	Current lift level 20 Current maximum Load 20000 Current Load 1800 End of lift Object 2.5 marks
61	Current lift level 10 Current maximum Load 10000 Current Load 1800 1.5 marks
62	End of lift Object 1 mark

(b)

```
void changeSpec(lift *x)
{   x->resetLift(20, 20000);
    x->display();
}
```

(c)

```
changeSpec(&l1);
```

Question 6 **[13 marks]**

Answer all questions in this section, based on the **Box** program in Program 2.0.

```
1   // Program 2.0
2   #include <iostream>
3   using namespace std;
4
5   class Box {
6      public:
7       // Constructor definition
8      Box(double l = 2.0, double b = 2.0, double h = 2.0)
9      { length = l;
10       breadth = b;
11       height = h;
12       cout <<"Initializing box with "
13            << length << breadth << height << endl;
14     }
15     double getVolume(
16     {    return length * breadth * height;  }
17
18     int compare(Box box)
19     {   return this->getVolume() == box.getVolume();  }
20
21          private:
22        double length;     // Length of a box
23        double breadth;    // Breadth of a box
24        double height;     // Height of a box
25  };
26
27  int main(void)
28  {
29     Box Box1(3.3, 1.2);          // Declare box1
30     Box Box2(1.2, 2.0, 3.3);     // Declare box2
31     Box Box3;                    // Declare box3
32
33     if (Box1.compare(Box2)) {
34        cout << "Box2 has the same volume with Box1" <<endl;
35     }
36     else
37        cout << "Box2 does not has the same volume with Box1"
38             << endl;
39     }
40     return 0;
41  }
```

a) Give the output produced by Program 2.0. [3 marks]

b) Rewrite `compare(Box box)` in line 18-19 to implement an overloaded operator **==** function. This function returns value true if the volume of the 2 boxes is the same.

[3 marks]

c) Rewrite Line 31 to implement the overload **==** function. [1 mark]

d) Declare an array named **`Boxes`** to hold 3 instances of Box and initialize the 3 instances of the array element similar to the instances of **`Box1`**, **`Box2`** and **`Box3`** in line 27-29.

[3 marks]

e) Based on an array declared in (c), use a loop to print the volume of the 3 boxes in the array.

[3 marks]

Question 7 **[15 Marks]**

Given the Program 4.0, answer the following question in this section.

```
 1  //Program 4.0
 2  #include <iostream>
 3  #include <string>
 4  using namespace std;
 5
 6  int main()
 7  {
 8          string w1 = "Have a nice day";
 9          string w2(3, "%");
10          string w3("Ali");
11
12          //Append a single character (,) to w1
13          w1 += ',';
14          cout << w1 << endl;
15
16          //Insert w2 into w1 before the word "day"
17          w2.insert(find("day"), w1);
18          cout << w1 << endl << w2 << endl;
19
20          //Append w3 to w1
21          w3.append(w1);
22          cout << w3 << endl << w1 << endl;
23
24          //Display the length of w1
25          cout << "The length of w1 = " << ________ << endl;
26
27          //Replace a substring "li" in w3 with "nuar"
28          ___________________
29          cout << w3 << endl;
30
31          return 0;
32  }
```

```
6(a) Initializing box with 3.3  1.2  2
     Initializing box with 1.2  2  3.3
     Initializing box with 2 2 2
     Box2 has the same volume with Box1
6(a) bool operator ==( Box &b) {
     if( getVolume() ==  b.getVolume() )
        return true;
     else
        return false;
     }
```

```
6(c) if (Box1 == Box2)
6(d) Box Boxes[3]={{3.3,1.2},{1.2,2.0,3.3}};
6(e) for (int i=0; i<3;i++)
       cout << "Volume : " << Boxes[i].getVolume()
            << endl;
```

a) Identify four statements with syntax and/ or logical errors (in **line 8** to **line 22**) based on the comment given in the program. Write your corrected statement in Table 2. (5 marks)

Table 2

Line	Corrected C++ Statement

b) Fill in the blanks with an appropriate function in C++ string class based on the comment given in the program. Write your answer in Table 3. (3 marks)

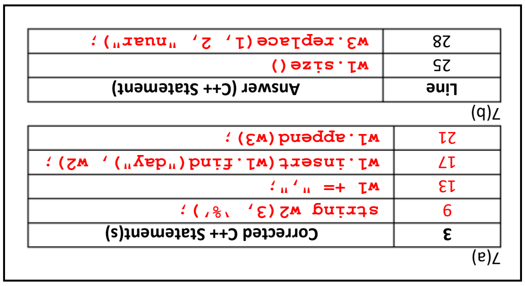
Table 3

Line	Answer (C++ Statement)
25	
28	

c) Assume all the errors are fixed in (a) and the program is completed in (b), what is the output of the program? Write your answer in Table 4. (7 marks)

Table 4

Line	Output
14	
18	
22	
25	
29	

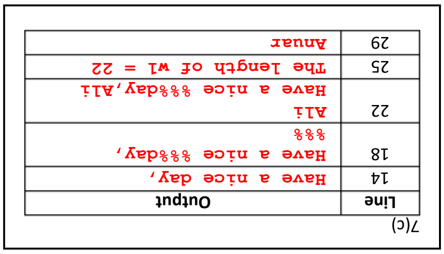


7(a)

3	Corrected C++ Statement(s)
9	`string w2(3, '%');`
13	`w1 += ",";`
17	`w1.insert(w1.find("day"), w2);`
21	`w1.append(w3);`

7(b)

Line	Answer (C++ Statement)
25	`w1.size()`
28	`w3.replace(1, 2, "nuar");`



7(c)

Line	Output
14	`Have a nice day,`

18	`Have a nice %%%day,`
	`%%%`

22	`Have a nice %%%day,Ali`
	`Ali`

25	`The length of w1 = 22`
29	`Anuar`