



UNIVERSITI TEKNOLOGI MALAYSIA

MID TERM TEST

SEMESTER II 2017/2018

SUBJECT CODE : SCSJ10123
SUBJECT NAME : PROGRAMMING TECHNIQUE II
YEAR/COURSE : 1 (SCSB/SCSJ /SCSP/SCSR / SCSV)
2 (SCSR/SCSV)
TIME : 9:00 AM – 11:00 AM (2 HOURS)
DATE : 30 MARCH 2018
VENUE : BK1-7, N28, FC

SOLUTIONS

SECTION A: STRUCTURED QUESTIONS

[60 MARKS]

Question 1

[15 Marks]

- a. (10 marks). Choose only FOUR (4) principles.

For each principle (2.5m): **description 1.5m + example 1m**

i. classes

A class is a category that generalize things with the same characteristics.

Example: Vehicle is a class which can represent for cars, lorries, and other vehicles.

Alternative:

A class is the template or mould or blueprint from which objects are made.

A class encapsulates the attributes and operations that characterize the objects.

ii. objects

An object is an exact instance from a class.

Example: Proton Saga with the registration number of JQS1234 is a specific car which is a type of car.

Alternative:

Classes can be used to instantiate as many objects as are needed.

Each object that is created from a class is called an instance of the class.

A program is simply a collection of objects that interact with each other to accomplish a goal.

iii. Encapsulation

Encapsulation is a key concept in working with objects: Combining attributes and methods in one package and hiding the implementation of the data from the user of the object.

Example: _Class Circle wraps the attributes radius and center for a circle with some other operations like the operation to calculate area, to calculate the circumference and so on.

iv. Data Hiding

Data hiding ensures methods should not directly access instance attributes in a class other than their own. Programs should interact with object attributes only through the object's methods.

Example: _Class Circle contains several attributes like radius and center for a circle. However, other entities (i.e., outsiders) cannot see the attributes. Thus, to do operations on the attributes, the outsiders need to do it via the dedicated operations (such as mutators and accessors) of the class.

v. Associations

Association: relates classes to each other through their objects.

Association can be, one to one, one to many, many to one, or many to many relationships.

Example: _Lectures teach courses

vi. Inheritance

Inheritance is the ability of one class to extend the capabilities of another.

it allows code defined in one class to be reused in other classes

Example: Class Insect is an inheritance of (or is a type of) class Animal

vii. Polymorphism

Polymorphism is the ability of objects performing the same actions differently.

Example:_ All insects have the same ability to move from one point to another. However, each insect performs their movement differently, such as grasshoppers move by jumping while ants by crawling.

b. (5 marks).

0.5 m each (including correct notation such as – or + and data type).

Employee
-name : string -salaryRate : double
+Employee() +Employee(n:string, rate:double) +setName(n:string) : void +setSalaryRate(r: double) : void +getName() : string +getSalaryRate() : double +getSalary(totalHour:int) : double

Question 2

[25 Marks]

a.

i. (4 marks)

Line	Output (4 marks. 0.5 each line)
54	value contains 100 link points to 5
57	value contains 100 link points to 5
63	value contains 100 link points to 11
65	value contains 88 link points to 11

ii. (3 marks). 1m each

a = 100	b = 100	c = 95
---------	---------	--------

iii. (3 marks). 1m each

Data with value 11 is being destroyed
Data with value 7 is being destroyed
Data with value 1 is being destroyed

b.

```
#include<iostream>
using namespace std;

class Rectangle{
private:
    int x1,y1;    // bottom-left corner
    int x2,y2;    // top-right corner

public:

    // The default constructor
    Rectangle(){ x1=y1=x2=y2=0; }

    // The constructor that sets the bottom-left and top-right corners
    Rectangle(int a, int b, int c, int d){
        x1=a; y1=b;
        x2=c; y2=d;
    }

    // (i). Define a constructor to create a rectangle with the bottom-left corner (0,0) and the
    // width and height set as w and h, respectively. ( 1 mark)

    Rectangle(int w, int h){    // 0.5m
        x1=y1=0; x2=w; y2= h;    // 0.5m
    }

    // (ii). Define a constructor to create a square with the bottom-left corner (x, y),
    // and the side length of s. (1.5 marks)

    Rectangle(int x, int y, int s){    // 0.5m
        x1=x; y1=y;    // 0.5m
        x2 = x1 + s; y2 = y1 + s;    // 0.5m
    }

    // (iii). Define a member function (method) that determines the width of the rectangle.
    // (1.5 marks)

    int getWidth() const{    // 0.5m
        return x2 - x1;    // 1m
    }

    // (iv). Define a member function (method) that determines the height of the rectangle.
    // (1.5 marks)

    int getHeight() const{    // 0.5m
        return y2 - y1;    // 1m
    }

    friend void inputRectangle(Rectangle&);
}; // End of class
```

```

// (v). Write the definition code for the function inputRectangle to read the coordinates of a
// rectangle from the keyboard. (1.5 marks)

void inputRectangle(Rectangle &r) // 0.5m
{
    cout << "Enter the rectangle's corners ( x1 y1 x2 y2) => ";
    cin >> r.x1 >> r.y1 >> r.x2 >> r.y2; // 1m
}

int main()
{
    // (vi). Declare an array to hold 4 objects of Rectangle, and initialize the array as follow:
    // the first element is a rectangle with bottom-left and top-right corners of (1,2) and
    // (3,4), respectively.
    //
    // the second element is a rectangle with bottom-left corner at (0,0), and width is 5 and height
    // is 10 .
    //
    // the third element is a square with bottom-left corner at (3,4), and the side length is 5
    // (2 marks)

    Rectangle rects[4] = { Rectangle(1,2,3,4), // 2m
                           Rectangle(5,10),
                           Rectangle(3,4,5)
                           };

    // (vii). Read the data for the fourth element of the array from the keyboard. (1 mark)

    inputRectangle(rects[3]) ; // 1m

    // (viii). Using a loop, calculate the total area of the rectangles and print the result. (5 marks)

    int sumArea = 0; // 0.5m
    for (int i=0; i<4; i++){ // 1m
        int area = rects[i].getHeight()*rects[i].getWidth(); //1.5m
        sumArea += area; // 1m
    }

    cout << "Total area = " << sumArea << endl; // 1m

    return 0;
}

```

Question 3**[20 Marks]**

Question	Answer (C++ Statement(s))
(a) 1 mark	<code>fstream &inp</code>
(b) 1 mark	<i>Either one:</i> <code>!inp</code> <code>inp.fail() @</code> <code>inp == NULL</code>
(c) 3 marks	<code>int len = fileName.length(); // 1m</code> <code>for (int i=0; i<len; i++) // 1m</code> <code>fileName[i] = tolower(fileName[i]); // 1m</code>
(d) 2 marks	<code>fin.open(fileName.c_str(), // 1m</code> <code>ios::in ios::binary // 1m</code> <code>);</code>
(e) 2 marks	<code>Car::fileCheck(fin);</code>
(f) 3 marks	<code>fin.seekg(0,ios::end); // 1.5m</code> <code>fileSize = fin.tellg(); // 1.5m</code>
(g) 1 mark	<code>objectSize = sizeof(Car);</code>
(h) 1 mark	<code>nCars = fileSize / objectSize;</code>
(i) 4 marks	<code>fin.seekg(0, ios::beg); // 1.5m</code> <code>fin.read((char *) (cars), nCars * objectSize); // 2.5m</code>
(j) 2 marks	<code>for (int i = 0; i < nCars; i++) // 1m</code> <code>totalPrice += cars[i].getPrice(); // 1m</code>

SECTION B: PROGRAMMING QUESTION**TOTAL – 40 MARKS**

Solution to this part is given in a separate file.