SIT102 – Introduction to Programming

Answers for 9.1P: Responses to Test 2

Student Name: yizheng he
Student ID: 221411294

Test 2 questions are released online via Unit Site >> Assessment >> Quizzes >> Test 2 (Week 9 PASS task) since week 9. Make sure you have read the task sheet 9.1P carefully for all details e.g. test duration, and question types.

**This answer sheet requires you to provide the following work:**

A. The **screenshot** of the auto-marked score of your Test 2 **with your name and student ID on it**
    - You can get back the score for those auto-marked questions, your questions set and your responses via Unit Site >> Assessment >> Quizzes. Click the downward arrow next to Test 2 and choose Submissions.
    - If needed – more 90 mins were spent on Test 2, a reflection with 50-100 words.

B. Your **revision** (revised answer after your self-review) of **each question with any sub-part** that you haven't scored their mark in the auto-mark section
    - as a form of the self-review on your own work with a fix by you.
    - If you attain 100% score, you don't have to do any revision for this part.

C. The **screenshot** of the hand execution **questions** that was assigned to you.
    - The hand execution data set is assigned by the unit site quiz tool. We need your help in providing the screenshot of your hand execution **questions version** to us so that we could mark your hand execution process.

D. Your **hand execution process** answers
    - **If** you get the **hand execution process related questions correct** in your auto-marked result, directly provide your work-out that has been finished during the Test time to the answer sheet. Your tutor will review it.
    - **Else**, provide your **revised hand execution process** into the answer sheet. Your tutor will review your revision.

E. Your **UML diagram** answer
    - Provide your work-out that has been finished during the Test time to the answer sheet. Your tutor will review it.

After that, save this answer sheet with your answers as a pdf file and upload it to OnTrack 9.1P.

Note: The PDF file of your 9.1P answer sheet is the only file you have to upload for OnTrack 9.1P.

A. The Screenshot of the Auto-marked Score of your Test 2 *with your name and student ID on it*

Guideline 1/2: The score for auto-mark sections will be available in unit site >> Assessment >> Quizzes. Click the downward arrow next to Test 2 and select Submissions.
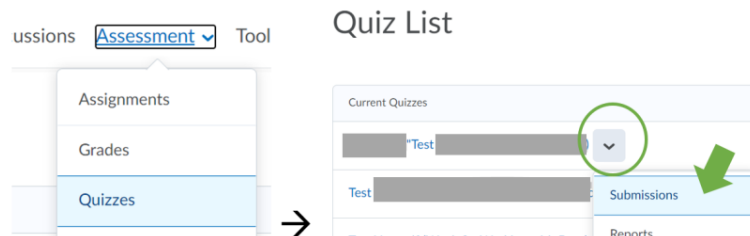
Guideline 2/2: Then, you can reach this the following page with your auto-marked score of your Test 2

Screenshot your auto-marked score and paste it in the following box:

Question: **If** your Test 2 unit site submission has been marked as late by the unit site quiz system, give your reflective writing here in 50 - 100 words to address it regarding:
- how much extra time than the expected 90 mins have you spent in the unit site Test 2? and
- which questions and topics took most of your time in the test, why is it so?

If you have finished and submitted your Test 2 to the unit site in the timed 90 mins, you don't have to deliver this reflection – just leave this answer box blank.

B. Your revision (revised answer after your self-review) of each question that you haven't scored a mark in the auto-mark section

- If you attain 100% score, you don't have to do any revision for this part – Just leave it blank.

1. Revised answer(s) for the question(s) and its sub-part(s) that you haven't scored their mark in the auto-mark section

- For example, if you scored 3/5 for a question with 5 sub-parts (a)-(e), you have to self-review your own responses for the entire question and provide your revision for the two missed sub-parts. Your tutor will review your revision.

2. Since different students will have different datasets in the questions, therefore also screenshot your question and paste it in the following box along with your revision.

Given the following program with two enum data types,

//code begins

```
#include "splashkit.h"

enum control
{
    OFF,
    ON
};

enum light
{
    WHITE = 1,
    RED,
    YELLOW,
    GREEN,
    PURPLE
};

int main()
{
    light color;
    control button;

    int temp_prev = 42;
    int temp_curr = 44;
    button = OFF;

    if ( (temp_curr > temp_prev and temp_curr > 44) or !button)
    {
        color = RED;
        write(color);
        color = GREEN;
        write(color);
        color = YELLOW;
        write(color);
        color = PURPLE;
        write(color);
    }
    else
    {
        color = WHITE;
        write(color);
        color = WHITE;
        write(color);
        color = PURPLE;
        write(color);
        color = PURPLE;
        write(color);
    }

    return 0;
}
```

//code ends

1a) What will be the output to the Terminal when the above program has been successfully executed? Type your answer into the answer box provided.

___5432___

1b) What is the integral value stored in OFF of the enum *control*? Type your answer into the answer box provided.
__0___

1a) What will be the output to the Terminal when the above program has been successfully executed? Type your answer into the answer box provided.
Should be 2435. **Because, in the if condition button is Boolean, off is 0, not off is on=1(!button) hence we run the if loop ,** the output of the program is RED, GREEN, YELLOW, PURPLE.
Following the enum struct, when white=1,then red =2 and soon on.

This is a program written in Swift programming language:

//code begins

```
1    let widths = [5,7,8]
2    let heights = [6,4,4]
3
4    var dimension_1:Int
5    var dimension_2:Int
6    var result:Float
7
8    func multiply(a: Int, b: Int) -> Int
9    {
10       return a*b
11   }
12
13   func division(a: Int, b: Int) -> Float
14   {
15       return Float(a) / Float(b)
16       //In swift, both numerator and denominator should be of the same type
17   }
18
19   for i in 0...2
20   {
21       dimension_1 = widths[i]
22       dimension_2 = heights[i]
23       result = division (a: multiply(a: dimension_1, b: dimension_2), b: 2)
24
25       if result > 15.0
26       {
27           print (result)
28       }
29
30   }
```

//code ends

2a) What will be the printed output when the given Swift program has been run? Type your answer into the answer box provided.

    14

2b) Based on the given Swift program, how many input parameter(s) is/are accepted by the function *division*. Type your answer as an integer into the answer box provided.

    3

2a. Starting from nineteen line, starting from 0-2, dimension_1 represents width[i] and dimension_2 represents height[i]. Result = division(a: multiply(a:dimension_1, b:dimension_2),b:2) 5*6/2=15, 7*4/2=14, 8*4/2=16.
When result>15.0, print(result), the output is 16.
2b. In division, this function takes two parametres. division(a,b) , then have 2 input parameter by the function division

This question 8 requires you to (i) draw a UML diagram and (ii) address 8(a) to 8(e) questions - by fill in the blanks with C++ codes.

- Draw a UML diagram (in the style covered in week 7) to **represent the data entities model of all custom data types** in the given program. You are required to paste your UML diagram to OnTrack 9.1P answer sheet. Your OnTrack tutor will evaluate your work via your 9.1P submission.
- Analyse the information of a C++ program in Figure 1, Figure 2, and Figure 3. Indicate your syntax for parts (a) to (e) respectively to complete the program (read Figure 3 for the given code). Syntax for parts (a) to (e) will be auto-marked by the system, remember to input accurate identifiers and code.

Figure 1 shows the sample output of the program. The program aims to read in and store data, then display the full list of customer names in the database to the Terminal.

Figure 1: The program's sample output

```
===== Collecting information =====
Enter store name: Skyview
***** Collecting information *****
Enter customer's name: Deakin
Enter customer's id: 1
Enter customer's age: 1
Enter the purchase amount:123.4
***** Collecting information *****
Enter customer's name: Glory
Enter customer's id: 2
Enter customer's age: 18
Enter the purchase amount:432.1
===== Full list of all customers in the database: =====
1 : Deakin
2 : Glory
```

Figure 2 shows read_ family's (functions *read_string(...)*, *read_integer(...)*, and *read_double(...)*) declarations.

Figure 2: read_ family's declarations

```
/**
 * read_ family reading inputs in various types
 * and returns the desired data in the correspondng data type
 * //Codes are hidden in this question
 */
> string read_string(string prompt)...

> int read_integer(string prompt)...

> double read_double(string prompt)...
```

Figure 3 shows parts of the program source code with comments, which includes:

- a struct identified as *customer_data*
- a struct identified as *store_data*
- a struct identified as *brand_kingdom*
- a function *gather_information()*
- a procedure *add_information(...)*
- a procedure *write_information(...)*
- a function *main()*

Figure 3: The C++ syntax for the above-mentioned program components

//code extract begins

```
1    struct customer_data
2    {
3      string name;          //store a name
4      int id;               //store the unique customer id
5      int age;              //store an age
6      double spent_amount;  //store the amount of purchase
7    };
8
9    struct store_data
10   {
11     string name;          //store a name
12     string location;      //store a location
13   };
14
15   struct brand_kingdom
16   {
17     store_data store;
18     vector<customer_data> customers;
19   };
20
21   // function to gather information for customer_data
22
23   // You are asked to complete the implementation of gather_information()
24   customer_data gather_information()
25   {
26     customer_data new_customer; // Declare a variable to store the read-in data
27
28     write_line("***** Collecting information ***** ");
29     new_customer.name = read_string("Enter customer's name: ");
30     new_customer.id = read_integer("Enter customer's id: ");
31     new_customer.age = read_integer("Enter customer's age: ");
32     new_customer.spent_amount = read_double("Enter the purchase amount:");
33
34     return    (a)    ; // return the collected data
35   }
36
37   // procedure to add information into the brand_kingdom
38   // brand_kingdom member is passed by reference with both read and write privileges
39   // You are asked to complete the implementation of add_information(...)
```

```
40   void add_information(brand_kingdom &head)
41   {
42       write_line("===== Collecting information =====");
43       //get the store's name and save in brand_kingdom
44       [ (b) ] = read_string("Enter store name: ");
45
46       // Declare a variable to store the read-in data for customer_data
47       customer_data customer;
48       // To collect two customers' data
49       int num = 2;
50       for (int i = 0; i < num; i++)
51       {
52           //call gather_information() and store the returned output
53           [ (c) ] = gather_information();
54           //add customer_data to the vector using push_back()
55           [ (d) ] .push_back(customer);
56       }
57   }


58
59   // procedure to print out all collected customer names to terminal
60   // brand_kingdom member is passed by reference with only read privilege
61   // You are asked to complete the implementation of write_information(...)
62   void write_information(const brand_kingdom &head)
63   {
64       write_line("===== Full list of all customers in the database: =====");
65       for (int i = 0; i < head.customers.size(); i++)
66       {
67           write(i+1);
68           //print out the customer name in the database
69           write_line(" : " + [ (e) ] );
70       }
71   }
72
73   int main()
74   {
75       brand_kingdom head;
76       add_information(head);
77       write_information(head);
78
79       return 0;
80   }
```

//code extract ends

Q8
A: new_customer;
Inside the function gather_information(), a variable named new_customer of type customer_data is created and we are inputting the data into each of the customer_data members and returning the new_customer data.
B: head.store.name.
We are reading the store name into the head struct which contains store struct and vector of customer.
C: customer.
we are calling gather_information() and returning the output into the customer variable.
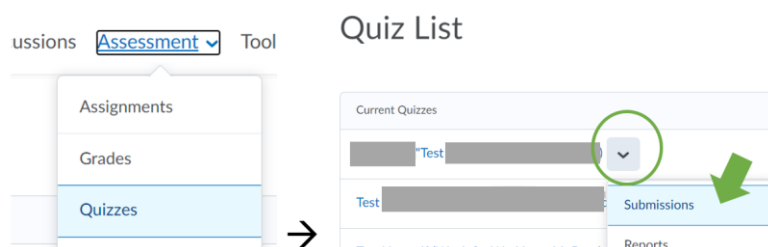D: head.customers
Here we are adding the customer to the vector customers which is a member of struct variable head
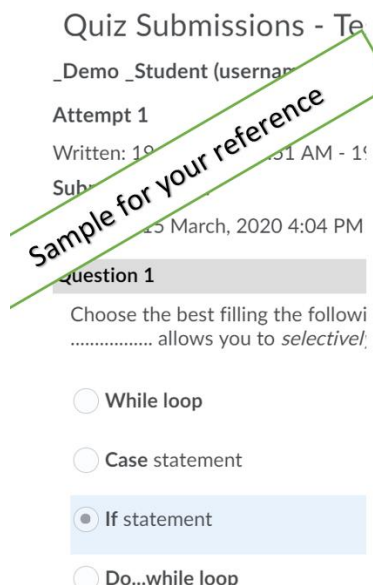E: main.customers[i].name
Here customers vector contains customer data of each head

C. The Screenshot of Hand Execution Questions assigned to you

Guideline 1/2: Go to unit site >> Assessment >> Quizzes again. Click the downward arrow next to this Test 2 and choose Submissions.

Quiz List

Assessment ∨	Tool
Assignments	
Grades	
Quizzes	

Current Quizzes

Test

Test

Submissions

Reports

→

Guideline 2/2: Click Attempt 1 to get back your questions and responses record.

Quiz Submissions - Te

_Demo _Student (usernam

Attempt 1

Written: 1

1 AM - 1

Sub

Sample for your reference

5 March, 2020 4:04 PM

Question 1

Choose the best filling the followi
…………… allows you to *selectivel*

○ While loop

○ Case statement

◉ If statement

○ Do...while loop

Screenshot(s) of your Hand Execution Questions assigned to you

5) Hand execute the data flow for **ALL** available variables based on the given code extract on your paper/drawing software, **and** then submit the **value** of **input[2]** at the end of the code to the answer box provided. *System auto-marking will be applied to the numerical answer in this question.*

Hint: *i--* is equivalent to *i = i - 1 (The value of i is decreased by 1)*

After this timed Test 2, your hand execution process/drawing answer is required to submit to OnTrack task 9.1P for tutor's review. Remember to keep your hand execution drawing(s).

//code extract begins

```
int size = 4 ;
int input [size];
for ( int i = size - 1 ;  i >= 0 ;  i -- )
{
    input [ i ] = 6 * i ;
}
```

//code extract ends

Answer:

12

6) Hand execute the data flow for **ALL** available variables based on the given code extract on your paper/drawing software, **and** then submit the **value** of **variable** *result* at the end of the code to the answer box provided. *System auto-marking will be applied to the numerical answer in this question.*

After this timed Test 2, your hand execution process/drawing answer is required to submit to OnTrack task 9.1P for tutor's review. Remember to keep your hand execution drawing(s).

//code extract begins

```
int result = 0;

vector<int> data;

data.push_back(7);
data.push_back(-6);
data.push_back(-5);
data.push_back(2);
data.push_back(13);

for (int j = 0; j < data.size(); j++ )
{
    if ( data[ j ] > 2 )
    {
        result = result + data[ j ] + j ;
    }
}
return result;
```

//code extract ends

Answer:

24

7) Hand execute the data flow for **ALL** available variables based on the given code extract on your paper/drawing software, **and** then submit the **value** of **int_array[1]** at the end of the code to the answer box provided. *System auto-marking will be applied to the numerical answer in this question.*

After this timed Test 2, your hand execution process/drawing answer is required to submit to OnTrack task 9.1P for tutor's review. Remember to keep your hand execution drawing(s).

//code extract begins

```
    vector<int> int_array;

    int_array.push_back(125);
    int_array.push_back(230);
    int_array[0] = 105;
    int_array.pop_back( );
    int_array.push_back(195);
    int_array.push_back(70);
    int_array.push_back(20);
    int_array.pop_back( );

    for ( int count = 2; count < int_array.size(); count++ )
    {
        write_line(count);
        int_array[count] = 350;
    }
```

//code extract ends

Answer:

195

D. Your answer for Hand Execution Process

1. **If** you get the hand execution process related questions **correct** in your auto-marked result, directly provide your work-out that has been finished during the Test time to the answer sheet. Your tutor will review it.

2. **Else**, provide your revised hand execution process into the answer sheet. Your tutor will review your revision.

Image(s)/drawing(s) of your answers for the hand execution process

Size 4

i

| 3̶ 2̶ 1̶ 0̶ -1 |

	3	2	1	0
input	18	12	6	0

First, int result = 0 ~~and vector<int> data~~

After ~~to~~ the execution of all data.push_back operations

=> ~~Vector<int> data~~ vector<int> data

0	1	2	3	4
7	-6	-5	2	13

Loop

At j=0, data[j] = 7 => 7 is great than 2 => result = result + data[i] + j

= 0 + 7 + 0 = 7 => result = 7

At j=1, data[i] = -6 => -6 is less than 2 => result was no change

At j=2, data[i] = -5 => -6 is less ~~than~~ than 2 => result was no change

At j=3, data[i] = 2 => 2 is equal 2 not great than 2 => result was no change

At i=4 data[i] = 13 => 13 is great than 2 => result + data[i] + j

= 7 + 4 + 13 = 24 => result = 24

End Loop

Hence, result = ~~7~~ 24

D	1	2	3	4
7	-6	-5	2	13

Data

~~size 4~~.   size ~~4~~ 5

j   ~~0 1 2 3 4~~ 5

result   ~~0 0 7~~ 24

G7

int-array

count   ~~2~~ 3
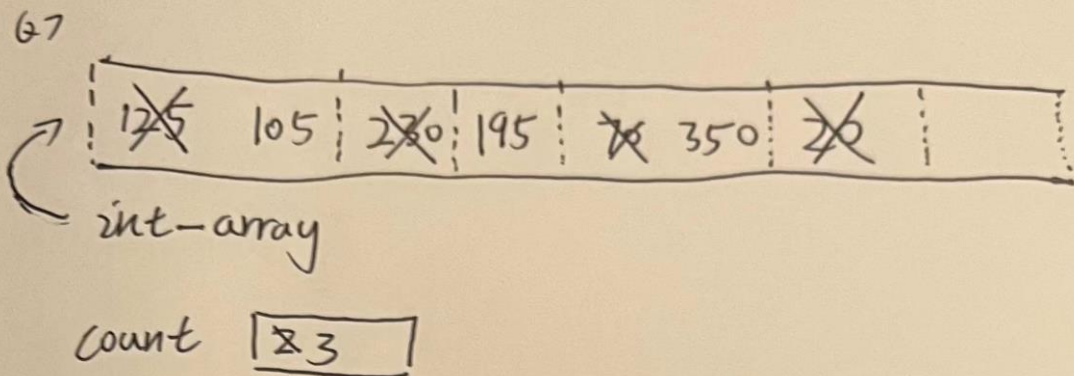
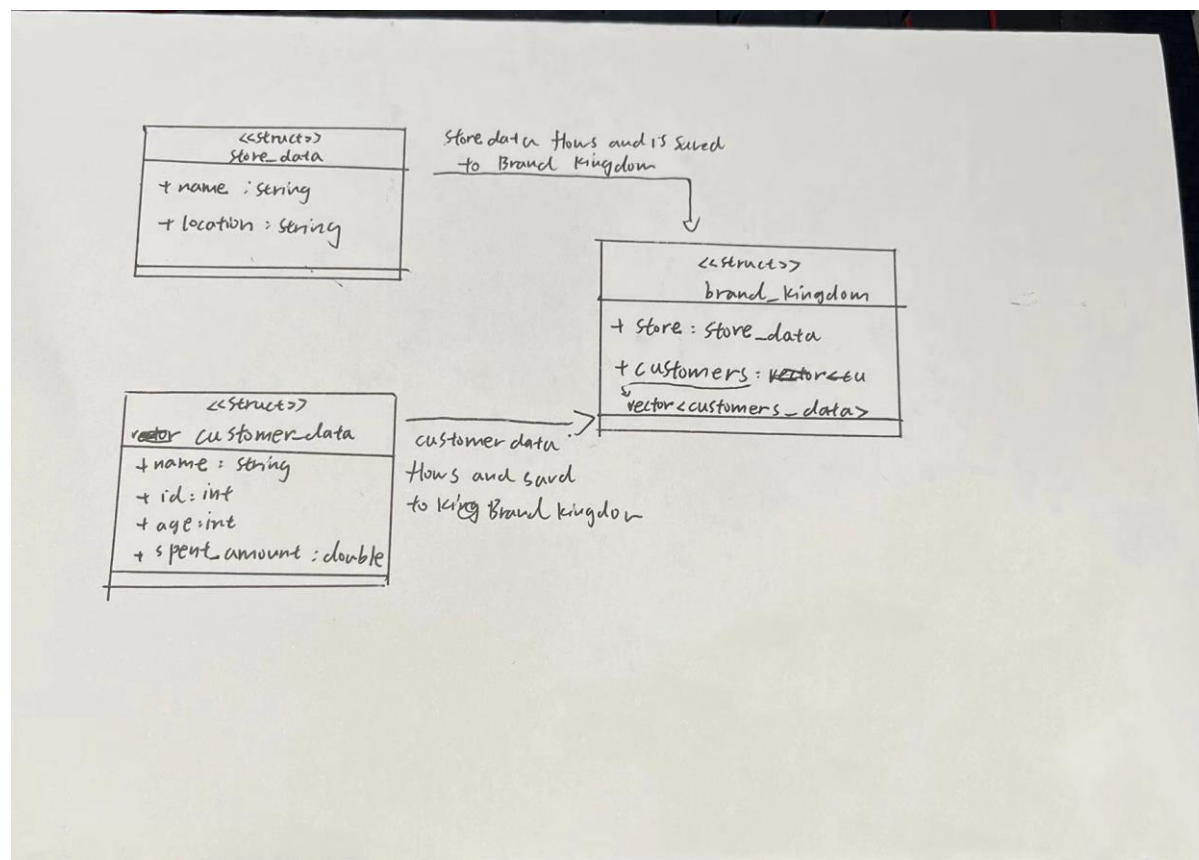## E. Your answer for UML diagram

1. Provide your UML diagram work-out that has been finished during the test time in this part. Your tutor will review it.

Image(s)/drawing(s) of your answer for the required UML diagram

- End of 9.1P answer sheet -