University of Kelaniya – Sri Lanka

Centre for Distance & Continuing Education

Bachelor of Science (General) External

First year Second semester examination - 2019

(New Syllabus)

2023 March

Faculty of Science

Computer Science

COSC 17543 – Object Oriented Programming

No. of Questions: **Four (04)** No. of pages: **Three (03)** Time: **Two (02) hours**

Answer <u>All</u> Questions

1.

(a) Explain one key feature for each of the following languages:

(i) Machine Language

(ii) Assembly Languages

(iii)High Level Languages

(b) List two (02) characteristics of Procedure Oriented Programming (POP) languages.

(c) Briefly explain two (02) advantages of Object Oriented Programming (OOP) languages.

(d) Briefly explain four pillars (concepts) of Object Oriented Programming (OOP).

(e) Identify suitable classes, their attributes and methods of a student management system.

(f) Identify the superclass, subclass(es) and draw the inheritance hierarchy for following scenarios:

(i) If *Class A* inherits from *Class B.*

(ii) *Animal*, *Insect*, and *Mammal* classes.

(iii) *Vehicle*, *Automobile*, *Motorcycle*, *Sports Car*, *Sedan (4-door car)*, and *Bicycle*.

2.

(a) State whether each of the following statements is *true* or *false*. If your answer is *false*, give reasons.

(i) Java considers the word ***Class*** as a key word.

(ii) Java identifier must begin with a ***letter***, _ or ***$***.

(iii) *int* and *long* datatypes in Java allows to store integer and floating-point values respectively.

(iv) The '+' operator can be used to combine two strings in an output statement in Java.

(b) Evaluate the following expressions.

(i) 3 * 3 + 3 % 2

(ii) (3+5) (float) 7

(iii) 3 + 2 / 5 + -2 * 4

(iv) 2 * (1 + -(3/4) / 2) * (2 - 6 % 3)

(c) Define a Java class called *Account* according to the following specifications:

(i) *Account* is represented by *account number*, *account name*, *account type*, *balance* attributes.

(ii) Add default constructor to the class which assign *0* to *account number*, "*unassigned*" to *account name*, "*unassigned*" to *account type* and *0* to *balance*.

(iii) Add four accessor methods to access the *account number*, *account name*, *account type*, *balance*.

(iv) Add four setter methods to set the *account number*, *account name*, *account type*, *balance*.

(v) Add two additional methods called *deposit* and *withdrawal* to add and deduct money from the account balance.

(d) Write a code fragment to declare and create two *Account* objects from the above prat (c) account class and name them *acct1* and *acct2*. Set the balance to 300,000.00 and 500.000.00, respectively. Set the name of owner for both accounts to "*UoK*".

(e) Write a code fragment to declare and create another *Account* object from the above prat (c) account class and name it *acct3*. Then, assign *acct2* object in part (d) to *acct3* object.

(f) Draw the state-of-memory diagram for the above *acct1*, *acct2* and *acct3* objects.

3.

(a) Write a Java conditional statement that implements the table below, where *grade* is an integer and *school* is a string. Both variables have been declared and *grade* has been initialized to a value greater than or equal to 1:

If *grade* is:	Set school to:
1,2,3,4,5	Elementary School
6,7,8	Junior School
9,10,11,12	High School
> 12	College

(b) Write a Java **switch** statement for the table above that is identical to the statement in the previous problem.

(c) Evaluate the following boolean expressions. For each of the following expressions, assume x is 10, y is 20, and z is 30. Indicate which of the following boolean expressions are always true and which are always false, regardless of the values for x, y, or z.

(i) x < 10 || x > 10

(ii) x < 10 && x > 10

(iii) x > y || y > x

(iv) !(x < y + z) || !(x + 10 <= 20)

(d) Write a Java program that reads employee's number of hours worked and the wage per hour. Then, it calculates the weekly wage of an employee. If the employee has worked for more than 40 hours, then he or she gets twice the wages per hour, for every extra hour that he or she has worked.

4.

(a) What will be the value of *sum* after the following loops are executed?

(i)

```
int count = 0, sum = 0;
    while ( count < 10 ) {
    sum += count;
    count++;
}
```

(ii)

```
int count = 0, sum = 0;
    while ( count < 20 ) {
    sum += 3*count;
    count += 2;
}
```

(iii)

```
sum = 0;
j = 0;
    do {
        j++;
        for (int i = 5; i > j; i--)
            sum = sum + (i+j);
    } while (j < 11);
```

(iv)

```
sum = 0;
for (int i = 0; i <= 5; i++)
    for (int j = 0; j <= 5; j++)
        sum += i ;
```

(b) Declare an array of double of size 365 to store daily temperatures for one year. Using this data structure, write a code fragment to implement the following tasks:

(i) Display the temperature values for the January month.

(ii) Find the average of daily temperature for the given year.

(iii) Find hottest and coldest days of the year.

(iv) The temperature of any given day. The day is specified by two input values: month (1, . . ., 12) and day (1, . . . , 31). Reject invalid input values (e.g., 13 for month and 32 for day).

(v) The difference between the hottest and coldest days of every month.

****