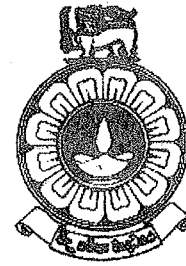


The Open University of Sri Lanka
Faculty of Natural Sciences
B.Sc./ B. Ed Degree Programme



Department	: Computer Science
Level	: 04
Name of the Examination	: Final Examination
Course Code – Title	: CSU4616 - OBJECT ORIENTED PROGRAMMING USING C++ AND JAVA
Academic Year	: 2024/2025
Date	: 28.11.2024
Time	: 01.30 p.m. – 04.30 p.m.
Duration	: Three hours only

General Instructions

1. Read all instructions carefully before answering the questions.
2. This question paper consists of **SIX** questions in **EIGHT** pages.
3. Answer **ANY FOUR** questions, in the provided answer book.
4. Answer to each question should commence from a **new page**.
5. Involvement in any activity that is considered as an exam offense will lead to punishment.
7. Use **blue or black ink** to answer the questions.
8. Clearly state your **Registration Number** for B.Sc/B.Ed Degree Programme in your answer script.

QUESTION 01

(a) Answer each of the following using Java programming language.

- (i) Declare the variables **age**, **height** and **gender** such that **age** is an integer, **height** is a decimal and **gender** is a character.
- (ii) Prompt the user to enter an integer **reg_no** and a string **name**. Then print the statement “My name is, and my registration number is,”, filling the blanks with values entered by the user.

- (iii) Declare a variable *answer*, then compute the result of dividing the integer *p* by the integer *q* and store the result in *answer*.
- (iv) Rewrite the statement in (iii) by handling the possibility of an error occurring if *q* is zero, using exception handling techniques.
- (b)
- (i) What is **Object Oriented Programming (OOP)**?
- (ii) How does OOP differ from Procedure Oriented Programming? Explain briefly using **three** key differences.
- (c) Clearly explain the differences of the following terms by giving suitable examples for each.
- (i) Class method and Instance method
- (ii) Nested class and Inner class
- (iii) Static Binding and Dynamic Binding
- (iv) Constructors and Methods
- (d)
- (i) Define the purpose of **access modifiers** in OOP languages.
- (ii) Copy the following table in your answer book and complete it to show the access levels for the four access modifiers in Java.

Visibility				
From the same class				
From any class in the same package				
From any class outside the package				
From a sub-class in the same package				
From a sub-class outside the same package				

- (e) State whether the following statements are True or False with respect to C++. If the statement is FALSE correct it and justify your answer.
- (i) `int &Marks` is a valid variable declaration.
- (ii) `if (x = 5)` is a valid way to check if *x* is equal to 5.

(iii) The following is a valid method declaration.

```
Public void setName(char Name){  
    Myname = name;  
}
```

(iv) The following loop is valid and will execute 5 times:

```
for (int i = 0; i < 5; i++){  
    {  
        cout << i << endl;  
    }  
}
```

[25 Marks]

QUESTION 02

(a) Write down **four** advantages of Java programming language that makes java a powerful programming language compared to languages such as C++.

(b)

(i) What is **inheritance**? Explain with a suitable example.

(ii) Provide a brief description of **five** different types of inheritance. Use relevant diagrams for each.

(c) In Java, the 'super' keyword allows access to members of the superclass for various purposes. State **three** different purposes of using super keyword.

(d) Answer the following questions based on the program segment given below.

```
public class Mother{  
    public String name;  
    public int age;  
    mother(){  
        name = "Nayana";  
        age = 45;  
    }  
    public void display(){  
        System.out.println("Name: " + name);  
        System.out.println("Age: " + age);  
    }  
}
```

- (i) Create a class named `Child` that inherits the `Mother` class. Declare member variables `name` and `Type` for the class `Child`. The type of the child may be a string which can take either son or daughter.
- (ii) Define a constructor to the class `Child` and give appropriate initial values for the instances `name`, `age` and `Type`.
- (iii) Define a method `display()` in the class `Child`, it should call the `display()` method in the superclass using the keyword `super`. The method `display()` in the class `Child` should print the information such as `name`, `age` and `type`.
- (iv) Create another class named `Main` for the main method and create **two** objects for the two classes (`Mother` and `Child`). Then execute the `display()` method for `Mother` and `Child`.

Example of the Output:

```
Mother's Information:
Name: Nayana
Age: 45
Child's Information:
Name: Damith
Age: 15
Type: Son
```

[25 Marks]

QUESTION 03

- (a) Imagine you are developing a computer game using Java programming language. Your game includes animals. Different types of animals make unique sounds. The program requirements specify that each animal should have its own sound behavior, and some animals should be able to make sounds repeatedly based on a given number. Follow the steps below to create the necessary classes and methods in Java
- (b)
 - (i) Create a base class named `Animal` with a method `sound()`. This method should print "Some generic animal sound". This serves as a default sound for any animal.
 - (ii) Create a specific animal class, `Dog` that inherit from the `Animal` class. In this `Dog` class override the `sound()` method to print "Bark".
 - (iii) Create another specific animal class, `Cat` that inherit from the `Animal` class. In this `Cat` class override the `sound()` method to print "Meow".

(iv) In the Dog class, add an overloaded version of the `sound ()` method with an integer parameter representing the number of barks.

E.g.,

Calling `dog.sound(3)` ; should output "Bark Bark Bark".

This demonstrates method overloading in the Dog class.

(v) In the main method,

(1) Create an instance of Dog called "Tommy" and call the `sound ()` method.

(2) Create another instance of Cat called "Kitty" and call the `sound ()` method.

(3) Directly call the overloaded `sound(int times)` method on the Dog instance

(c) Briefly explain the following terms by providing suitable examples for each.

(i) Class

(ii) Object

(d) Write a C++ code to create an object of Student called `myStudent`.

[25 Marks]

QUESTION 04

(a) Consider the following C++ code segment.

Briefly explain the purpose of having `friend void xyz(void) ;` statement as in line number 08.

```
01class ABC
02{
03    -----
04    -----
05    public:
06        -----
07        -----
08        friend void xyz(void) ;
09};
```

(b) Briefly explain the following terms with suitable examples.

(i) Aggregation

(ii) Association

(iii) Composition

- (c)
- (i) Briefly describe the concept of **thread** and **multi-threaded programming**.
 - (ii) Explain **two** benefits of using multithreading in applications.
- (d) Explain the purpose of an **Inner class** by providing **three** advantages.
- (e)
- (i) *"Interface is different from a class."*
Assess the validity of this statement by providing **four** key differences.
 - (ii) Following Java code segment shows the interface that represents a simple currency converter, from and to Sri Lankan Rupees (SLR).
- ```
public interface CurrencyConverter{
 float fromSLR(float amount);
 float toSLR(float amount);
}
```
- Create a class called `DollarCurrencyConverter` for supporting the conversion between US dollars and Sri Lankan rupees. You can assume that the rate of US dollar is 290.

[25 Marks]

### QUESTION 05

- (a) Define a class in JAVA to represent a **Point** with the following data fields and methods. Use access modifiers and data types appropriately.
- (i) Data fields – `x_coordinate` and `y_coordinate`
  - (ii) Parameterized constructor to assign the values that are passing during object creation
  - (iii) Selector method for two member variables
  - (iv) Modifier methods for two member variables
  - (v) Write a main class called `Test` to test the `Point` class. Create an object and call all the methods defined in the `Point` class.
- (b) What is an **Abstract Method**?
- (c) How do you achieve the data abstraction of a program? Illustrate it by redefining the `Point` class appropriately. (No need to write the whole class again).
- (d) Briefly explain the following terms in terms of C++.
- (i) Constructor
  - (ii) Destructor

- (e) Consider the following C++ code, which has two functions that add numbers of different types.

```
int plusFuncInt(int x, int y) {
 return x + y;
}

double plusFuncDouble(double x, double y) {
 return x + y;
}

int main() {
 int myNum1 = plusFuncInt(8, 5);
 double myNum2 = plusFuncDouble(4.3, 6.26);
 cout << "Int: " << myNum1 << "\n";
 cout << "Double: " << myNum2;
 return 0;
}
```

Instead of defining two functions that should do the same thing, it is better to overload one. This process is called *function overloading*.

Rewrite the above code in terms of *function overloading*. Use `plusFunc` as the function name.

[25 Marks]

### QUESTION 06

*A software developer wants to build an attendance system that can automatically detect faces in images to mark attendance. To do this, the developer plans to use Java-based face detection software. The developer does not have access to the software's source code but has the Java bytecode and some documentation.*

- (a) Explain each of the following terms:

- Source Code
- Machine Code
- Java Bytecode

- (b) List **two** advantages of distributing Java software as bytecode instead of source code.

(c) Fill in the blanks using appropriate Java terms from the list given below.

(Hiding, private, final, static, protection, this)

- (i) Abstraction and encapsulation are two forms of data \_\_\_\_\_.
- (ii) The keyword \_\_\_\_\_ is used as a modifier for a class to prevent inherited to another class.
- (iii) The keyword \_\_\_\_\_ is used to refer to the current object and it can be used anywhere as a reference to an object.
- (iv) The keyword \_\_\_\_\_ is used to declare a class variable within a class, outside any method.

(d)

(i) Explain **Exception** and **Exception Handling** in brief.

(ii) Give **two** runtime exceptions with examples.

(e) Consider the following C++ code.

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

void m2(int x) {
 cout << 4 << endl;
 if (x == 1)
 throw runtime_error("IO error");
 if (x == 0)
 throw overflow_error("Arithmetic error");
 cout << 5 << endl;
}

void m(int x) {
 try {
 m2(x);
 cout << 1 << endl;
 } catch (overflow_error &e) {
 cout << 2 << endl;
 } catch (runtime_error &e) {
 cout << 3 << endl;
 }
}
```

(i) What is the output of calling m(1).

(ii) What is the output of calling m(0).

[25 Marks]

\*\*\* All Right Reserved. \*\*\*