

THE OPEN UNIVERSITY OF SRI LANKA
B.Sc. DEGREE PROGRAMME : LEVEL 04
DEPARTMENT OF COMPUTER SCIENCE
FINAL EXAMINATION – 2017/2018
CPU2242/CSU4616: OBJECT ORIENTED PROGRAMMING USING
C++ AND JAVA
DURATION: Three Hours (3 Hours)



Date: 20.09.2018

Time : 1.30 pm -4.30 pm

Answer FOUR Questions ONLY.

Q1)

- a) "Object Oriented Programming treats data as a critical element when compared to Procedure Oriented Programming". Explain the above statement by describing data handling in Procedure Oriented Programming and Object-Oriented Programming.
- b) List three (3) advantages of JAVA over C++ in World Wide Web (www).
- c)
 - i. Define **abstract class** and **abstract method** by giving an example per each.
 - ii. What is the difference between an abstract class and a normal class?
- d) Define a class in **Java** to represent a **Book** with following data members and methods
 - i. Data members --title, ISBN and author.
 - ii. A parameterized constructor to initialize the data members of the class.
 - iii Selector and modifier methods for Book ISBN member variable.
 - iv A method to print the Book details. (Title, ISBN, Author)
- e) Write **C++** statements for the following.
 - i. Create an automatic object called Book1 from class Book and call the print method.
 - ii Create a dynamic object called Book2 from class Book and call the print method.

Q2)

- a)
 - i. Define the term **Object** and **Class**. Explain these concepts using a suitable example.
 - ii. What is the difference between Object and Class.
- b) “Java is a platform independent language”. Explain the platform independency related to JAVA language.
- c) Fill in the blanks using the appropriate term from the given list.
(private, public, protected)
 - i. Amember is accessible only to the class in which it is defined.
 - ii All super class members declared as and are inherited by the subclass.
 - iii All members declared protected in a superclass become.....members in subclasses, they cannot be
- d) Define a class in C++ to represent a **Student** with the following data members and methods.
 - i Private data members – Student name, Student ID
 - ii. Default constructor with default values
Student name-“Nimal”, Student ID-“S0035”
 - iii A user defined constructor and a copy constructor.
 - iv Destructor
 - v Selector and modifier methods for Student ID member variable (they should be inline methods)
 - vi Write a suitable main method to test the defined methods in part (d- i, ii, iii, iv and v)
- e) Explain constructor and destructor indicating their propose to be defined in a class. Provide examples from the class defined in part(d).

Q3)

- a)
 - i What is inheritance?
 - ii State two different forms of inheritance supported by C++. Explain with an example.
- b)
 - i Define a class named **Point** in C++ to represent a **2D Point** with **X** and **Y** coordinates. The class has the following methods.
 - A user defined constructor
 - A method to print the X and Y values.
 - ii Create a subclass called **Circle** inherited from **Point** class which contains following properties.
 - A member variable to store the radius of the circle
 - A user defined constructor to initialize X, Y and radius
 - Override the print method
- c) How **Java** refers the hidden variables, Constructors and Overridden methods in the superclass from the subclass. Explain with suitable Java statements.
- d) State whether the following statements are **TRUE** or **FALSE** with respect to **Java**. If the statement is **FALSE** correct it by explaining the reason.
 - i A subclass inherits all members including constructors in its superclass.
 - ii An abstract method should be implemented in the derived class.
 - iii **this** keyword acts as the assignment operator to assign argument values to the member variables.
 - iv All the instances of a class share the same copy of the Class variable.
 - v An abstract class can be instantiated.

Q4)

- a) What is the purpose of using **Final** keyword when declaring variables, methods and classes? Explain briefly by providing examples for each case.
- b)
 - i Define a class in **Java** named as **BankAccount** to represent the details of a bank account. It has a two attributes called account number, balance and the following methods.
 - A user defined constructor

- A method to deposit an amount to the bank account
 - A method to withdraw an amount from the bank account
 - A method to display the bank account details
- ii Define a class named as **SavingsAccount** inherited from the class bank account.
- A user defined constructor
 - Override the withdraw method. Customer must maintain a minimum balance of Rs 10000/= in a savings account
 - Override the display method
- c) Write a main class called **Test**. Create an object of Savings Account class and call all the methods belong to that class.
- d) Typically, banks provide several types of accounts. Redefine the BankAccount class as an abstract class. Define one method appropriately as an abstract method.

Q5)

- a) i Explain **Polymorphism** in the context of Object-Oriented Programming.
- ii What is meant by **overriding**? Discuss your answer using method (function) prototypes.
- b) Define a class named **Complex** in C++ to represent a complex number which includes real and imaginary parts in floating point values. Include the following member to the class.
- i. A user defined constructor
 - ii To overload + operator to add two complex numbers.
(Example : $(3 + 2i) + (2 + 2i) = (5 + 4i)$)
 - iii To overload == operator to check whether two complex numbers are equivalent.
(Example : two complex numbers $(x + yi)$ and $(a + bi)$ are equivalent if, the real parts are equal $(x=a)$ and the imaginary parts are equal $(y=b)$)
 - iv To overload << operator to print a complex number.
- c) Write a suitable main method to test the defined functions in part (c- i, ii, iii, iv)
- d) Explain operator overloading, taking examples from class, Complex in part(b)

Q6)

- a) Explain the following terms in brief by giving suitable examples.
 - i. Encapsulation
 - ii. Aggregation
 - iii. Logical errors
 - iv. Runtime errors

- b) Define a class in **Java** called **Calculator** with the following methods using method overloading concept.
 - i. A method to add two integer numbers.
 - ii A method to add three integer numbers.
 - iii A method to add two float numbers.
 - iv A method to subtract two integers.

- c) Write a main class called **Test**. Create an object of Calculator class and call all methods belong to that class. Print the results as follows.

23 + 34 = 57
2+4+6 = 12
2.7+3.2=5.9
29-5= 24

- d) Explain method overloading, taking method signatures as examples from the class Calculator created in part(b)

- e) What is an Exception?

*** All Rights Reserved ***