

Object Oriented Programming

Time : Three Hours

Maximum Marks : 70

Note: i) Answer any five questions.

किन्हीं पाँच प्रश्नों को हल कीजिए।

ii) All questions carry equal marks.

सभी प्रश्न के समान अंक हैं।

iii) In case of any doubt or dispute the English version question should be treated as final.

किसी भी प्रकार के संदेह अथवा विवाद की स्थिति में अंग्रेजी भाषा के प्रश्न को अंतिम माना जायेगा।

1. a) Why do we use do-while loop in C? Also, tell any properties which you know?

हम C में do-while loop का उपयोग क्यों करते हैं? साथ ही, कोई ऐसी properties बताइए जिसे आप जानते हैं?

b) Write a program to check the number is palindrome or not. The program should accept any arbitrary number typed by user.

संख्या पैलिंड्रोम है या नहीं यह जाँचने के लिए एक प्रोग्राम लिखें। कार्यक्रम को उपयोगकर्ता द्वारा टाइप की गई किसी भी मनमानी संख्या को स्वीकार करना चाहिए।

2. a) Differentiate between Structure and Array with example.
उदाहरण के साथ स्ट्रक्चर और ऐरे के बीच अंतर करें।

b) Write a program in C++ to differentiate between function overloading and function overriding.

फंक्शन ओवरलोडिंग और फंक्शन ओवरराइडिंग के बीच अंतर करने के लिए C++ में एक प्रोग्राम लिखें।

3. a) Distinguish between data abstraction and data encapsulation using suitable code.

उपयुक्त कोड का उपयोग करके डाटा एब्सट्रैक्शन और डाटा एन्कैप्सुलेशन के बीच अंतर करें।

b) What is OOPs? And Describe OOPs concepts in brief.

OOPs क्या है? और OOP की अवधारणाओं का संक्षेप में वर्णन करें।

4. a) What are different types of Inheritance?

विभिन्न प्रकार के Inheritance क्या हैं?

b) Explain the basic concepts of Object Oriented Programming. Explain control structure with suitable example.

ऑब्जेक्ट ओरिएंटेड प्रोग्रामिंग की बुनियादी अवधारणाओं की व्याख्या करें। नियंत्रण संरचना को उपयुक्त उदाहरण सहित समझाइए।

5. a) Write briefly:

i) Data hiding

ii) Encapsulation

iii) Data Abstraction

संक्षेप में लिखें।

- i) डाटा हिडींग
- ii) एनकैप्सुलेशन
- iii) डाटा एब्सट्रैक्शन

- b) Distinguish between Abstract class and Interface with example.

उदाहरण के साथ एब्सट्रैक्ट क्लास और इंटरफेस के बीच अंतर करें।

6. a) Explain the difference between how method parameters are passed for variables that contain object references and variables that contain primitive data types?

ऑब्जेक्ट रेफरेंस वाले वेरिएबल के लिए मेथड पैरामीटर कैसे पास किए जाते हैं और वेरिएबल जिनमें आदिम डाटा टाइप होते हैं, के बीच अंतर स्पष्ट करें।

- b) Explain the difference between implementing an interface and a derived class.

एक इंटरफेस और एक व्युत्पन्न वर्ग को लागू करने के बीच अंतर स्पष्ट करें।

7. a) What is Template? Explain with the help of example how to create a Function Template.

टेम्पलेट क्या है? उदाहरण की सहायता से समझाइए कि फंक्शन टेम्पलेट कैसे बनाया जाता है?

- b) What are files and how these are handled in C++?

फाइलें क्या हैं और इन्हें C++ में कैसे नियंत्रित किया जाता है?

8. Write short note on any two of the following:

- i) Inline function
- ii) Auto storage class and static storage class
- iii) Virtual Functions

Total No. of Questions : 8]

[Total No. of Printed Pages : 6

Roll No. 0205 IT 223 D 03

IT-304 (GS)

B.Tech., III Semester

Examination, November 2022

Grading System (GS)

Object Oriented Programming and Methodology

Time : Three Hours

Maximum Marks : 70

- Note:** i) Attempt any five questions.
किन्हीं पाँच प्रश्नों को हल कीजिए।
- ii) All questions carry equal marks.
सभी प्रश्नों के समान अंक हैं।
- iii) In case of any doubt or dispute the English version question should be treated as final.
किसी भी प्रकार के संदेह अथवा विवाद की स्थिति में अंग्रेजी भाषा के प्रश्न को अंतिम माना जायेगा।

1. a) What is inheritance? What is the importance of inheritance? Differentiate between multilevel and multiple inheritance with suitable example.

इनहेरिटेंस क्या है? इनहेरिटेंस का क्या महत्व है? उपयुक्त उदाहरण की सहायता से मल्टीलेवल और मल्टीपल इनहेरिटेंस में अंतर स्पष्ट कीजिए।

- b) How is Data Encapsulation different from Information Hiding? Explain with example.

सूचना छिपाने से डाटा एनकैप्सुलेशन कैसे अलग है? उदाहरण सहित समझाइए।

PTO

2. a) What is the difference between overloading and overriding? How is binding used in polymorphism? Explain with example.

ओवरलोडिंग और ओवरराइडिंग के बीच क्या अंतर है? बहुरूपता में बाध्यकारी कैसे प्रयोग किया जाता है? उदाहरण सहित समझाइए।

b) What are the concepts or techniques used in object-oriented concept to handling complexity of system?

सिस्टम की जटिलता को संभालने के लिए ऑब्जेक्ट-ओरिएंटेड कॉन्सेप्ट में कौन-सी कॉन्सेप्ट या तकनीक का इस्तेमाल किया जाता है?

3. a) Write C++ function to overloaded versions of a Max function which compares two integers, two reals, or two strings, and returns the 'larger' one.

मैक्स फंक्शन के ओवरलोडेड संस्करणों के लिए C++ फंक्शन लिखें जो दो पूर्णांक, दो वास्तविक, या दो तारों की तुलना करता है, और बड़ा लौटाता है?

b) What is friend function? Explain application of friend function with the suitable example?

फ्रेंड फंक्शन क्या है? उपयुक्त उदाहरण के साथ फ्रेंड फंक्शन के अनुप्रयोग को समझाइए।

4. a) Write C++ programs to validation of the examination marks is to be done. The range of exam marks is between 65 and 100 (both inclusive). This is to be done by including a validate method in Student class.

C++ प्रोग्राम लिखने के लिए परीक्षा के अंकों का सत्यापन किया जाना है। परीक्षा के अंकों की सीमा 65 और 100 (दोनों सम्मिलित) के बीच है। यह विद्यार्थी वर्ग में एक मान्य पद्धति को शामिल करके किया जाना है।

Student class

Student
.....
.....
+validateExamMarks() : Boolean

b) Write a class in C++ which stores the marks of 3 subjects of a student. Provide methods to set the marks and get the values of the marks. Also, provide a method called getResult(), which prints the grade of the students. Grade is calculated as shown in the table given below :

C++ में एक कक्षा लिखिए जो एक छात्र के 3 विषयों के अंकों को संग्रहित करती है। अंक निर्धारित करने और अंकों के मान प्राप्त करने के तरीके प्रदान करें। साथ ही, getResult(), नामक एक विधि प्रदान करें, जो छात्रों के ग्रेड को प्रिंट करती है। ग्रेड की गणना नीचे दी गई तालिका में दर्शाई गई है।

Average of 3 subjects	Grade
80 to 100	A
65 to 79	B
50 to 64	C
40 to 49	D
0 to 39	E

5. a) We had several Branch Id's related to the various branches of engineering in the college. The fees to be paid are different for different branches. Write C++ programs to calculate the fees based on branch by using an array.

हमारे पास कॉलेज में इंजीनियरिंग की विभिन्न शाखाओं से संबंधित कई शाखा पहचान पत्र थे। भुगतान की जाने वाली फीस अलग-अलग शाखाओं के लिए अलग-अलग है। एक सारणी का उपयोग करके शाखा के आधार पर शुल्क की गणना करने के लिए C++ प्रोग्राम लिखें।

- b) Define a class named Complex for representing complex numbers. A complex number has the general form $a + ib$, where a is the real part and b is the imaginary part (i stands for imaginary). Complex arithmetic rules are as follows:

$$(a + ib) + (c + id) = (a + c) + i(b + d)$$

$$(a + ib) - (c + id) = (a + c) - i(b + d)$$

$$(a + ib) * (c + id) = (ac - bd) + i(bc + ad)$$

Define these operations as member functions of Complex.

सम्मिश्र संख्याओं का प्रतिनिधित्व करने के लिए सम्मिश्र नामक वर्ग को परिभाषित करें। एक सम्मिश्र संख्या का सामान्य रूप $a + ib$ होता है, जहाँ a वास्तविक भाग होता है और b काल्पनिक भाग होता है (i का अर्थ काल्पनिक होता है) जटिल अंकगणितीय नियम इस प्रकार हैं।

$$(a + ib) + (c + id) = (a + c) + i(b + d)$$

$$(a + ib) - (c + id) = (a + c) - i(b + d)$$

$$(a + ib) * (c + id) = (ac - bd) + i(bc + ad)$$

इन परिचालनों को कॉम्प्लेक्स के सदस्य कार्यों के रूप में परिभाषित करें।

6. a) Write a class Survey Operator derived from Person. This class has the following additional members:

Data Members:

NumberOfCalls, NumberOfSuccessfulCalls, TotalTime, Available

Operations:

GetNumberOfCalls, GetNumberOfSuccessfulCalls, SetAvailable, MakeCall

Define appropriate constructors for this class. Remember the fact that the base class Person does not have a default constructor and needs an explicit invocation using super().

व्यक्ति से प्राप्त एक वर्ग सर्वे ऑपरेटर लिखें। इस वर्ग में निम्नलिखित अतिरिक्त सदस्य हैं।

Data Members:

NumberOfCalls, NumberOfSuccessfulCalls, TotalTime, Available

Operations:

GetNumberOfCalls, GetNumberOfSuccessfulCalls, SetAvailable, MakeCall

इस वर्ग के लिए उपयुक्त कंस्ट्रक्टर्स को परिभाषित करें। इस तथ्य को याद रखें कि बेस क्लास व्यक्ति के पास डिफॉल्ट कंस्ट्रक्टर नहीं है और super() का उपयोग करके स्पष्ट आमंत्रण की आवश्यकता है।

- b) Create a class called Employee whose objects are records for an employee. This class will be derived class of the class Person which you will have to copy into a file of your own and compile. An employee record has an employee's name (inherited from the class Person), an annual salary represented as a single value of type double, a year the employee started work as a single value of type int and a national insurance number, which is a value of type String. Your class should have a reasonable number of constructors and accessor methods, as well as an equals method. Write a main() method to fully test your class definition.

कर्मचारी नामक एक वर्ग बनाइए जिसकी वस्तुएँ कर्मचारी के लिए रिकॉर्ड हो। यह क्लास पर्सन क्लास की व्युत्पन्न क्लास होगी जिसे आपको अपनी खुद की फाइल में कॉपी करना होगा और कंपाइल करना होगा। एक कर्मचारी रिकॉर्ड में एक कर्मचारी का नाम होता है। (वर्ग व्यक्ति से विरासत में मिला), एक वार्षिक वेतन टाइप डबल के एकल मूल्य के रूप में दर्शाया जाता है, जिस वर्ष कर्मचारी ने टाइप int के एकल मूल्य और एक राष्ट्रीय बीमा संख्या के रूप में काम करना शुरू किया, जो एक मूल्य है स्ट्रिंग प्रकार का।

आपकी कक्षा में उचित संख्या में कंस्ट्रक्टर और एक्सेसर विधियों के साथ-साथ एक समान विधि होनी चाहिए। अपनी कक्षा परिभाषा का पूर्ण परीक्षण करने के लिए एक main() विधि लिखें।

7. a) Write a C++ program to initialize base class members through a derived class constructor.

व्युत्पन्न वर्ग निर्माता के माध्यम से आधार वर्ग के सदस्यों को प्रारंभ करने के लिए C++ प्रोग्राम लिखें।

- b) What is the ambiguity that arises in multiple inheritance? How it can be overcome. Explain with example.

एकाधिक वंशानुक्रम में उत्पन्न होने वाली अस्पष्टता क्या है? इसे कैसे दूर किया जा सकता है? उदाहरण सहित समझाइए।

8. Create a base class, Telephone, and derive a class ElectronicPhone from it. In Telephone, create a protected string member phonetype, and a public method Ring() that outputs a text message like this: "Ringing the <phonetype>." In ElectronicPhone, the constructor should set the phonetype to "Digital." In the Run() method, call Ring() on the ElectronicPhone to test the inheritance. For above specification to illustrate a polymorphic method. Have the derived class override the Ring() method to display a different message.

एक बेस क्लास, टेलीफोन बनाइए और उससे एक क्लास इलेक्ट्रॉनिक फोन प्राप्त करें। टेलीफोन में, एक संरक्षित स्ट्रिंग सदस्य फोन प्रकार और एक सार्वजनिक विधि Ring() बनाइए जो इस तरह का पाठ संदेश देता है "Ringing the <phonetype>." इलेक्ट्रॉनिक फोन में, निर्माता को फोन प्रकार को "Digital." पर सेट करना चाहिए। Run() विधि में, इनहेरिटेंस का परीक्षण करने के लिए इलेक्ट्रॉनिक फोन पर Ring() को कॉल करें। उपरोक्त विनिर्देश के लिए बहुरूपी विधि का वर्णन करने के लिए। व्युत्पन्न वर्ग को एक अलग संदेश प्रदर्शित करने के लिए Ring() विधि को ओवरराइट करने दें।
