C++ and Multicore Programming

Unit I Foundations of Object Oriented Programming

Introduction to procedural, modular, object-oriented and generic programming techniques, Limitations of procedural programming, Need of object-oriented programming, fundamentals of object-oriented programming: objects, classes, data members, methods, messages, data encapsulation, data abstraction and information hiding, inheritance, polymorphism,

++: Extensions to C: Variable declarations, global scope, ‘const’, reference variables, comments, default parameters, function prototypes, function overloading, inline functions, default and constant arguments, ‘cin’, ‘cout’, formatting and I/O manipulators, new and delete operators Defining a class, data members and methods, public, private and protected members, inline member functions, static data members, static member functions, ‘this’ pointer, constructors, destructors, friend function, dynamic memory allocation, array of objects, pointers and classes, class as ADTs and code reuse

Unit II Overloading and Inheritance

Introduction, Need of operator overloading, overloading the assignment, binary and unary operators, overloading using friends, rules for operator overloading, type conversions Concept and need, single inheritance, base and derived classes, friend classes, types of inheritance, hybrid inheritance, member access control, static class, multiple inheritance, ambiguity, virtual base class, polymorphism, virtual functions, pure virtual functions, abstract base class, virtual destructors, early and late binding, container classes

Unit III Templates and Exception Handling

Introduction, Templates: Function template and class template, function overloading vs. function templates, member function templates and template arguments, Introduction to Generic Programming: Introduction to Standard Template Library (STL), containers, iterators and algorithms, study of container template classes for vectors and stacks and related algorithms, Namespaces: Introduction, Rules of namespaces

Exception Handling: Introduction, syntax for exception handling code: try-catch-throw, Multiple Exceptions, Exceptions with arguments, Introduction to RTTI
Managing Console I/O Operations: Introduction, C++ streams, stream classes, unformatted I/O, formatted I/O and I/O manipulators

Unit IV Challenges in Multi-core Programming

Sequential Models, Concurrency, Challenges for software development, New libraries for C++ developers, Processor Architecture Challenges, Operating systems (OS) roll in concurrent development: Consistent interfaces, Resource Management, OS interaction, Core OS services, Application Program Interfaces, Decomposition and Operating systems Roll, Hiding the Operating systems Roll: Abstraction and Encapsulation, Interface classes for POSIX API

Processes, Interface classes and predicates

Multicore and multiprocessors, Processes and threads, Parent-child relations, Process control block, Anatomy of a process, Process States, Process Scheduling, ps utility, process priorities, Context switch, Activities in process creation, Process environment variables, using system() to Spawn Processes, Killing a process, Process Resources, Asynchronous and Synchronous Processes, the wait() function call, Predicates, Processes and Interface classes

Unit V Multithreading

Thread, User and Kernel level threads, Thread Context, Hardware Threads and software threads, Comparing threads to processes, setting thread attributes, The architecture of a Thread, Compiling and linking threaded programs, Creating threads, Managing threads, Thread Interface classes

Unit VI Communication and synchronization of Concurrent tasks

Communication and synchronization, Synchronizing concurrency, Thread Strategy approaches, Decomposition and Encapsulation of Work Case studies of concurrency models

1. Create a class named weather report that holds a daily weather report with data members day_of_month, hightemp, lowtemp, amount_rain and amount_snow. The constructor initializes the fields with default values: 99 for
day_of_month, 999 for highemp, -999 for low emp and 0 for amount_rain and amount_snow. Include a function that prompts the user and sets values for each field so that you can override the default values. Write a C++/Java/Python program that creates a monthly report.

Menu driven program with options to Enter data and Display report

2 A book shop maintains the inventory of books that are being sold at the shop. The list includes details such as author, title, price, publisher and stock position. Whenever a customer wants a book, the sales person inputs the title and author and the system searches the list and displays whether it is available or not. If it is not, an appropriate message is displayed. If it is, then the system displays the book details and requests for the number of copies required. If the requested copies are available, the total cost of the requested copies is displayed; otherwise the message “Required copies not in stock” is displayed.

Design a system using a class called books with suitable member functions and Constructors. Use new operator in constructors to allocate memory space required. Implement C++ program for the system.

3 Develop an object oriented program in C++ to create a database of the personnel information system containing the following information: Name, Date of Birth, Blood group, Height, Weight, Insurance Policy number, Contact address, telephone number, driving license no. etc Construct the database with suitable member functions for initializing and destroying the data viz constructor, default constructor, copy constructor, destructor, static member functions, friend class, this pointer, inline code and dynamic memory allocation operators-new and delete.

4 Design a C++ Class ‘Complex ‘ with data members for real and imaginary part. Provide default and parameterized constructors. Write a program to perform arithmetic operations of two complex numbers using operator overloading (using either member functions or friend functions).

5 Write a C++ program to perform String operations
   i. = Equality
   ii. == String Copy
   iii. + Concatenation
   iv. << To display a string
   v. >> To reverse a string
   vi. Function to determine whether a string is a palindrome
To find occurrence of a sub-string. Use Operator Overloading

6 Develop an object oriented program in C++ to create a database of the personnel information system containing the following information: Name, Date of Birth, Blood group, Height, Weight, Insurance Policy number, Contact address, telephone number, driving licence no. etc Construct the database with suitable member functions for initializing and destroying the data viz constructor, default constructor, copy constructor, destructor, static member functions , friend class, this pointer, inline code and dynamic memory allocation operators-new and delete.

7 Write a program in C++ using function template to read two matrices of different data types such as integers and floating point values and perform simple arithmetic operations on these matrices separately and display it.

8 Design a C++ base class consisting of the data members such as name of the student, roll number and subject. The derived class consists of the data members subject code, internal assessment and university examination marks. Construct a virtual base class for the item name of the student and roll number. The program should have the facilities.
   i) Build a master table ii) List a table iii) Insert a new entry iv) Delete old entry v) Edit an entry vi) Search for a record

9 Create a C++ class named Television that has data members to hold the model number and the screen size in inches, and the price. Member functions include overloaded insertion and extraction operators. If more than four digits are entered for the model, if the screen size is smaller than 12 or greater than 70 inches, or if the price is negative or over $5000 then throw an integer. Write a main() function that instantiates a television object, allows user to enter data and displays the data members .If an exception is caught, replace all the data member values with zero values.

Group B (any Two)

1 A ‘C’ program function having one IF-THEN-ELSE returns the truth-ness value (TRUE/FALSE) is to be replaced by overloading while porting it to C++. Use appropriate overloading to replace IF-THEN-ELSE. Demonstrate the functioning by using it in a class.

2 A ‘C’ program uses a structure to implement a circular linked list for maintaining the numbers in ascending order. New arrival of number increases
the size of circular linked list. This program is to be ported to C++ using appropriate C++ Data structures and programming. (In C++ avoid use of structure and IF-Then-Else or while/do-while etc.)

3 Implement C++/Java/Python program to create a base class called shape. Use this class to store two double type values that could be used to compute the area of figures. Derive two specific classes called function get_data() to initialize base class data members and another member function display_area() to compute and display the area of figures. Make classes to suit their requirements. Using these three classes, design a program that will accept dimension of a triangle or a rectangle interactively, and display the area.
Remember the two values given as input will be treated as lengths of two sides in the case of rectangles, and as base and height in the case of triangles, and used as follows:
Area of rectangle = x*y
Area of triangle = 1/2*x*y

4 Implement C++/Java/Python program to implement a base class consisting of the data members such as name of the student, roll number and subject. The derived class consists of the data members subject code ,internal assessment and university examination marks. The program should have the facilities. i) Build a master table ii) List a table iii) Insert a new entry iv) Delete old entry v) Edit an entry vi) Search for a record. Use virtual functions.

5 Implement C++/Java/Python program to write a class template to represent a generic vector. Include following member functions:
To create the vector.
To modify the value of a given element
To multiply by a scalar value
To display the vector in the form (10,20,30,…)

6 Implement C++/Java/Python program for bubble sort using function template

7 Refer the standard template library to use list container and using C++/Java implement following member functions of list class: empty, insert, merge, reverse, sort

8 Write a C++/Java program for the following:
1) A function to read two double type numbers from keyboard
2) A function to calculate the division of these two numbers
3) A try block to throw an exception when a wrong type of data is keyed in
4) A try block to detect and throw an exception if the condition “divide-by-zero” occurs
5) Appropriate catch block to handle the exceptions thrown

9 Write a C++/Java program for the following:
Create a class named Television that has data members to hold the model number and the screen size in inches, and the price. Member functions include overloaded insertion and extraction operators. If more than four digits are entered for the model, if the screen size is smaller than 12 or greater than 70 inches, or if the price is negative or over $5000 then throw an exception. Write a main() function that instantiates a television object, allows user to enter data and displays the data members. If an exception is caught, replace all the data member values with zero values.

10 Create employee bio-data using following classes i) Personal record ii) Professional record iii) Academic record Assume appropriate data members and member function to accept required data & print bio-data. Create bio-data using multiple inheritance using C++/Java/Python.

11 Using multi-core programming implement POSIX-spawn() function to create a process

12 Implement a simple interface class for a POSIX Process using multi-core Programming

13 Using multi-core programming implement a predicate class

14 Implement POSIX queue class that encapsulates the basic function such as open, send, receive, remove, close. Use multi-core programming

15 Implement POSIX semaphore using multi-core programming

16 Using multi-core programming, implement Mutex semaphore for : Initialization Request ownership Release ownership Try ownership Destruction

17 Using multi-core programming implement a thread interface class

18 Write a Object Oriented Program using C++/Java for passing command line arguments to the thread function, using the command line argument to determine the number of threads, Use multi-core programming
Group C (Advanced Assignments) Any One

1. Write a concurrent program to implement the Odd-Even Merge Sort. Effective use of Multicore Architecture Core 1 and Core 2 effectively is expected.

2. Write a concurrent program to implement the Dining philosophers problem. Effective use of Multicore Architecture is expected.

3. Write a concurrent program for Matrix Multiplication. Effective use of Multicore Architecture is expected.