12CS46

USN | | | | | | | | | | |

R. V. COLLEGE OF ENGINEERING

Autonomous Institution affiliated to VTU

IV Semester B. E. Examinations April/May-16

Computer Science and Engineering

OBJECT ORIENTED PROGRAMMING WITH C++

Time: 03 Hours ***Maximum Marks: 100***

Instructions to candidates:

1. Answer all questions from Part A. Part A questions should be answered in the first three pages of the answer book only.
2. Answer FIVE full questions from Part B.

PART A

1	1.1	Define polymorphism.	01

	1.2	What is the output of following code?:
		$\#\ include\ <iostream.h>$
		$void\ main\ ()$
		{
		$\quad int\ a = 10;$
		$\quad cout\ <<++a<<\ "\ "\ <<a--<<"\ "\ <<\ a++<<"\ "\ <<a;$
		}
			02

	1.3	Define function overriding. Justify whether the following functions are overloaded: $int\ add\ (int, float);$ $float\ add\ (float, int);$	02
	1.4	Mention any two operators which cannot be overloaded in $C++$.	01
	1.5	What is the difference between early binding and late binding?	01
	1.6	Why is a method usually defined outside class using scope resolution operator (::) even though it is possible to define within the class itself?	01
	1.7	Mention the type of "this" pointer? How does it get created?	02
	1.8	Write a generic function to swap two values of any type.	02

	1.9	Give the output of the following code:
		$\#\ include\ <iostream.h>$
		$int\ main\ ()$
		{
		$\quad int\ x = -1;$
		$\quad cout\ <<\ "Before\ try\backslash n";$
		$\quad try$
		$\quad$ {
		$\quad\quad cout\ <<\ "Inside\ try\backslash n";$
		$\quad\quad if\ (x < 0)$
		$\quad\quad$ {
		$\quad\quad\quad throw\ x;$
		$\quad\quad\quad cout\ <<\ "After\ throw\backslash n";$
		$\quad\quad$ }
		$\quad$ }	

		catch (*int x*)
		{
		cout << "*Exception caught*\n";
		}
		cout << "*After catch*\n";
		return 0;
		}
			02

	1.10	Mention any two advantages of using friend classes.	02

	1.11	Give the output of the following code considering a 32-bit machine:
		class A
		{
		int a ;
		static float b;
		}
		class B: *public A*
		{
		int c;
		public :
		char ch [10];
		}
		void main ( )
		{
		B obj ;
		cout << *sizeof*(*obj*);
		}
			02

	1.12	Define pure virtual function.	02

PART B

2	a	Distinguish between structure and class.	04
	b	Write a $C++$ program to create two classes BOX $(WIDTH, HEIGHT, DEPTH)$ and $BOXCOLOR$ $(COLOR)$ to find volume of a box where the properties of $BOXCOLOR$ are derived from BOX. Set the properties of base class (BOX) through derived class $(BOXCOLOR)$ constructor.	08
	c	Explain the mechanism used to remove ambiguity when two classes having same name are defined in same scope.	04
		OR	
3	a	What is the use of copy constructor?	04
	b	Write a $C++$ program to compare two persons based on their age. Compare function should be a friend function. Display the details (Name, age, address, gender) of the person who is elder.	08
	c	Create a class which has a function to convert temperature in Fahrenheit into Celsius.	04
4	a	Write a $C++$ program to accept two matrices and perform matrix multiplication.	08
	b	Illustrate the order of invocation of constructors and destructors when an object of derived class is created.	08
		OR	

5	a	What is the new-handler function? Explain the working of sel-new-handler.	06
	b	Write a program in $C++$ to add two complex numbers and display the result.	10
6	a	What is constructor overloading?	04
	b	Develop overloaded functions to find area of $CIRCLE, SQUARE$ and $TRIANGLE$ shape.	08
	c	What is function overriding?	04
		OR	
7	a	List the operators already overloaded in $C++$ which were used in C language for other purpose. Explain the need for operator overloading.	06
	b	Write a $C++$ program to concatenate two strings defined in String class using operator overloaded function '$+$'.	10
8	a	Write a $C++$ program to create template class for quick sort program to sort array of characters, integers and real numbers.	08
	b	Design a class Animal which has a pure virtual function $talk()$. Derive two classes Dog and Human and redefine the $talk()$ method to represent the talking way of Dog and Human.	08
		OR	
9	a	What is standard template library? Describe the different operations performed on STL stack.	08
	b	Explain the following functions with programming examples: i) $copy\_backward()$; ii) $count()$; iii) $find()$.	08
10	a	Explain in detail the C-style handling of error generating code.	08
	b	Illustrate with an example of using try, catch, and throw keywords in exception handling.	08
		OR	
11	a	With an appropriate example, explain the use of following functions: i) $setw()$; ii) $setprecision(int)$; iii) $setfill()$.	08
	b	Illustrate the classification of different file handling streams.	08

12CS44

USN | | | | | | | | | | |

R. V. COLLEGE OF ENGINEERING

Autonomous Institution affiliated to VTU

IV Semester B. E. Examinations April/May-16

Computer Science and Engineering

OPERATING SYSTEMS

Time: 03 Hours ***Maximum Marks: 100***

Instructions to candidates:

3. Answer all questions from Part A. Part A questions should be answered in the first three pages of the answer book only.
4. Answer FIVE full questions from Part B.

PART A

1	1.1	_____ increases CPU utilization by organizing jobs so that CPU always has one to execute.	01
	1.2	List the two common models of communication.	01
	1.3	Define context switching.	02
	1.4	_____ scheduler controls the degree of multi-programming.	01
	1.5	What is preemptive and non-preemptive scheduling?	02
	1.6	What are spinlocks?	01
	1.7	Multiple-process solution to critical section problem is known as _____.	01
	1.8	List any two advantages of monitors.	02
	1.9	What are four necessary conditions for deadlock to occur?	02
	1.10	Distinguish between internal and external fragmentation.	02
	1.11	Consider the logical address space of 8 pages of 1024 words each mapped onto physical memory of 32 frames. How many bits are in logical and physical address?	02
	1.12	What is Belady's anomaly?	01
	1.13	The disk space allocation method used by $MS-DOS$ is _____.	01
	1.14	What is the use of modify bit?	01

PART B

2	a	Explain the distinguishing feature of the following operating systems: i) Real Time System; ii) Time Sharing System; iii) Distributed System.	08
	b	What are the differences between symmetric and asymmetric multi-processor?	04
	c	List different types of system calls. Explain any two in brief.	04
		OR	
3	a	What is a process? With the help of state transition diagram, explain the various states of a process.	04

b With a neat diagram, explain the concept of virtual machines and also discuss the benefits of virtual machines. 08

c What is meant by cooperating process? List any four advantages of cooperating process. 04

4 a Consider the following set of processes with length of CPU burst time given in $msec$:

$Process$	$Arrival\ Time$	$Burst\ Time$	$Priority$
P_1	0	10	3
P_2	0	1	1
P_3	0	2	3
P_4	0	1	4
P_5	0	5	2

i) Draw Gantt chart for $FCFS, SJF$, non-preemptive priority and Round-Robin (time slice = $1ms$);

ii) Compute average waiting time and turnaround time for the algorithms mentioned above. 10

b With a neat diagram, explain multilevel queue and multilevel feedback queue scheduling. 06

OR

5 a Why is a Thread called LWP? Explain different threading issues. Bring out the concept of thread pool. 08

b Write a note on benefits of multi-thread programming. 04

c Distinguish between user-level and kernel-level threads. 04

6 a Write pseudo code for Test and Set and Swap. Prove that it satisfies mutual exclusion. 08

b Define Dining Philosophers problem and give a solution for the same using monitors. 08

OR

7 a Draw the resource allocation graph and wait-for-graph for the following situation and explain whether the system is in deadlock state:

i) Process P_1 is holding resource R_2 and is waiting for resource R_1.

ii) $P2$ is using R_1 and waiting for R_3, R_4, R_5

iii) P_3 is using R_4 and waiting for R_5

iv) P_4 is using R_5 and waiting for R_2

v) P_5 is using R_3. 08

b Consider the following snap shot of the system:

$Process$	$Allocation$			$Claim$			$Available$		
	A	B	C	A	B	C	A	B	C
P_0	0	1	0	7	5	3	3	3	2
P_1	2	0	0	3	2	2			
P_2	3	0	2	9	0	2			
P_3	2	1	1	2	2	2			
P_4	0	0	2	4	3	3			

		Answer the following questions using Banker's algorithm: i) What is the content of need matrix?; ii) Is the system in safe state or deadlock state? Give the sequence.	08
8	a	With an example, explain the concept of overlays.	06
	b	Explain the two techniques for structuring the page table with suitable figures.	10
		OR	
9	a	Consider the following page reference string: 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1 How many page faults would occur for the following page replacement algorithm assuming three and four frames in the main memory? All the page frames are initially empty so that first unique pages will all cost one fault each: i) $FIFO$ ii) LRU iii) Optimal. Which algorithm is most efficient?	10
	b	What is thrashing? Explain the two methods of overcoming thrashing.	06
10	a	Suppose a moving disk has 200 tracks numbered 0 – 199. The current request being serviced by the head is for track 50 and the previous request serviced was for track 45 and if the queue of requests are: $50, 45, 180, 35, 150, 5, 145, 51$ and 54, what is the total number of head movements needed to satisfy these requests using $SSTF, SCAN$ and $C-SCAN$ algorithms?	10
	b	Discuss briefly the three different methods of allocating space for frames.	06
		OR	
11	a	Explain the different schemes for designing logical structure of directory.	08
	b	Explain the access matrix model for implementing protection in a computer system.	08

12CS45 / 12IS45

USN | | | | | | | | | | |

R. V. COLLEGE OF ENGINEERING

Autonomous Institution affiliated to VTU

IV Semester B. E. Examinations April/May-16

Common to CSE / ISE

DESIGN AND ANALYSIS OF ALGORITHMS

Time: 03 Hours ***Maximum Marks: 100***

Instructions to candidates:

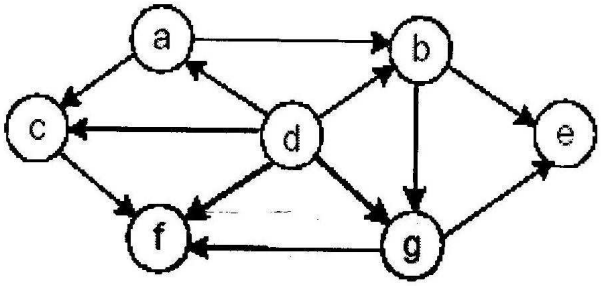
5. Answer all questions from Part A. Part A questions should be answered in the first three pages of the answer book only.
6. Answer FIVE full questions from Part B.

PART A

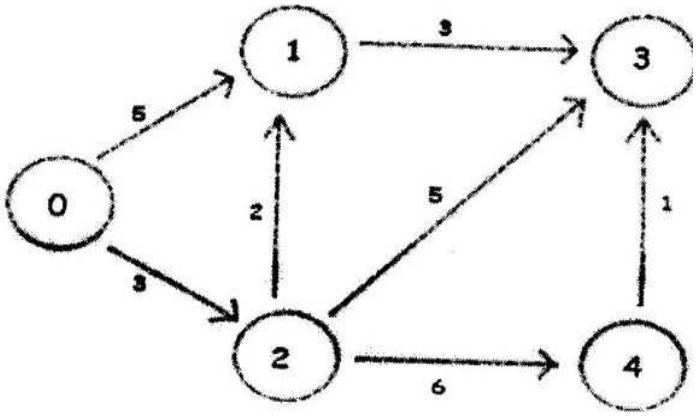
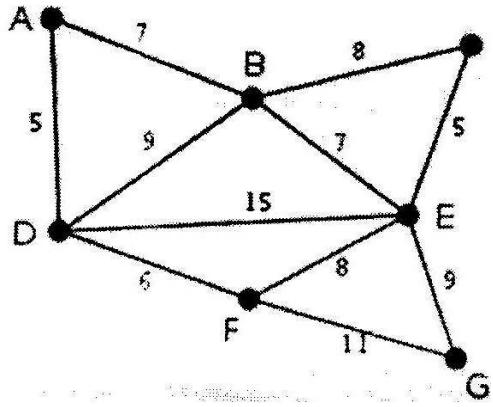
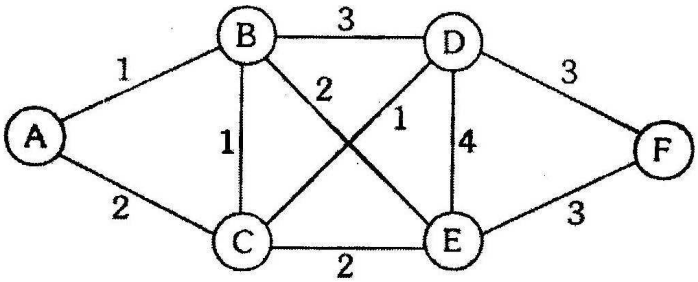
1	1.1	What is the total number of solutions one can get for 8-Queens problem?	01
	1.2	What is the time complexity of concatenating two doubly linked lists?	01
	1.3	What is articulation point?	01
	1.4	What is the time complexity of finding an item in assorted singly linked list?	01
	1.5	Define Hamiltonian circuit.	01
	1.6	Which algorithm design technique tends to be ideally suited for parallel computations?	01
	1.7	What is the time complexity of multiplying two matrices?	01
	1.8	Write the shift table and Good suffix table for the following pattern: $ABCBDAB$	01
	1.9	List the drawbacks of AVL trees.	02
	1.10	Consider the problem of finding $A \cap B$ (where A and B are two sets of numbers). What is the time complexity of solving the above problem by brute force and by presorting based approach?	02
	1.11	For the following list apply binary search and find out the largest number of key comparison: $19, 25, 27, 32, 42, 56, 66, 71, 73, 75, 82$	02
	1.12	Differentiate between BFS and DFS algorithms.	02
	1.13	Solve the recurrence relation by backward substitution method $T(n) = T(n-1) + 1$.	02
	1.14	Analyze the efficiency of Prims algorithm.	02

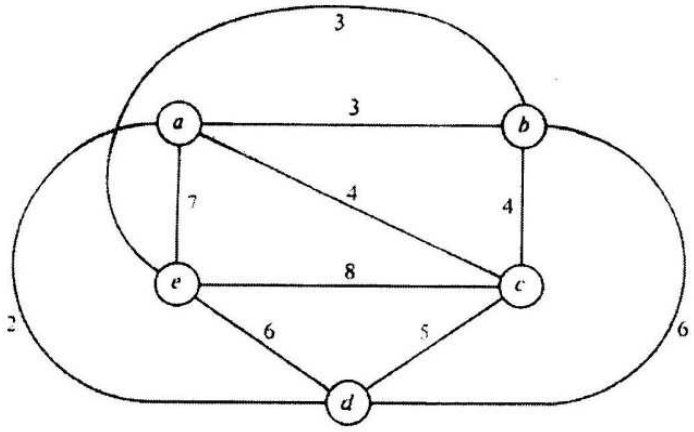
PART B

2	a	Explain recursive Euclidian algorithm for finding $GCD\ (m, n)$ and analyse its efficiency.	06
	b	Describe the usage of limits for comparing orders of growth. Compare $log_2 n$ and $(n)^{1/3}$	03
	c	Briefly discuss the framework for designing and analyzing an algorithm.	07
		OR	

3	a	Define big-oh, big-theta and big-omega notations.	06
	b	State Master theorem and evaluate $T(n) = 10T(n/3) + 18n^3$ by Master theorem method.	04
	c	Design and analyze recursive Tower of Hanoi problem.	06
4	a	Compute 1234×1234 using Divide and Conquer technique.	05
	b	Apply *DFS* method to solve topological ordering for the following graph. 	05
	c	Construct *AVL* tree for the following elements: $130, 120, 110, 100, 90, 80, 70, 60, 50, 40, 30, 20, 10, 5$	06
		OR	
5	a	Construct $2-3$ tree for the following elements: *UNCOPYRIGHTABLE*	05
	b	Consider the problem of finding k^{th} smallest element: i) Design a pre-sorting-based algorithm for solving this problem and determine its efficiency. ii) Design a Decrease and Conquer algorithm for solving this problem and determine its efficiency.	06
	c	Derive the worst case efficiency analysis of quick sort algorithm.	05
6	a	Apply Brute force string matching algorithm to find the pattern *SHIFT* in the following text: *ISE_DEPARTMENT_IS_SHIFTED* Write the worst case scenario (give pattern and text) for the above algorithm.	05
	b	Write the recursive function of 0/1 knapsack problem by dynamic programming method and solve the following with the same method. Maximum capacity-5	07
	c	Design an algorithm to compute $5C_2$ or $C(5,2)$ and analyze its efficiency.	04
		OR	
7	a	Apply Boyer-Moore algorithm to find the pattern *STING* in the following text: *A STRING SEARCHING EXAMPLE CONSISTING OF TEXT*	05

Item	*Value*	*Weight*
1	3	2
2	4	3
3	5	4
4	6	5

	b	Find all-pair shortest oath algorithm for the following graph:  	07
	c	Sort H, E, A, P, S, O, R, T in alphabetical order using heap sort.	04
8	a	What is spanning tree? Apply Kruskal's algorithm for the following graph and find minimum spanning tree. Discuss the application of the same.  	09
	b	Construct a Huffman tree for the following data and obtain its Huffman code: $Character$ | A | B | C | D $Probability$ | 0.1 | 0.4 | 0.2 | 0.3 Encode the text $BADDAA$ using the code of the question.	07
		OR	
9	a	Write Dijkstra's algorithm and apply the same for the following graph by taking 'A' as source. Analyze the efficiency of the algorithm.  	10
	b	Write the decision tree for 3-element insertion sort.	04
	c	Define Bipartite graph.	02

10	a	Find the optimal solution for the assignment problem given below by brute force method and by branch and bound method: $\begin{pmatrix} 9 & 5 & 4 & 5 \\ 4 & 3 & 5 & 6 \\ 3 & 1 & 3 & 2 \\ 2 & 4 & 2 & 6 \end{pmatrix}$	12
	b	Write the state-space tree to solve the following instance of the subset-sum problem: $S = \{2, 3, 4, 5\}$ and $d = 7$.	04
		OR	
11	a	Solve the following travelling salesman problem by branch and bound method and brute force method. 	10
	b	Define N, NP problems with an example.	04
	c	Analyze the efficiency of Knapsack problem by brute force approach.	02