

<p>Q. No. 1. What is the output of the following C program?</p> <pre>#include<stdio.h> struct XYZ { int a; struct XYZ *next; }; int main() { struct XYZ temp; temp.a = 10; temp.next = NULL; printf("%d", temp.a); return 0; }</pre> <p>A: 10 B: Garbage value C: Compile time error D: Runtime error</p> <p><input checked="" type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D Clear Answer Mark For Review</p>	<p>Explanation: struct XYZ temp:: This line declares a variable temp of type struct XYZ. temp.a = 10:: This line sets the integer member a of temp to 10. temp.next = NULL This line sets the pointer member next of temp to NULL, indicating that there is no next element in a potential linked list. printf("%d", temp.a); This line prints the value of the member a of temp.</p>
<p>Q. No. 2. What is the problem with the following C program code?</p>	
<p>D: Runtime error</p> <p><input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input checked="" type="radio"/> D Clear Answer Mark For Review</p> <p>Q. No. 2. What is the problem with the following C program code?</p> <pre>#include<stdio.h> #include <stdlib.h> int main() { int *p = (int *)malloc(sizeof(int)); int *g=p; free(p); *g=10; return(0); }</pre> <p>A: Results in dangling pointer B: Compile time error C: Results in memory leak D: Runtime error</p> <p><input checked="" type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D Clear Answer Mark For Review</p>	<p>Explanation: malloc allocates the memory and the p holds the address. Both p and g now point to the same dynamically allocated memory. freeing p pointer means p is no more pointing to the allocated memory address and as g was also pointing to same memory its also free now. so now both p and g are dangling pointer as the memory they were pointing does not exist now.</p>
<p>Q. No. 3. What is the output of the following C program?</p> <pre>#include<stdio.h> void g(int *x, int *y) { *y=x; *y = 3; } int a = 1, b = 2; int main() { g(&a, &b); printf("%d %d\n", a, b); return 0; }</pre> <p>A: 3 2 B: 3 1 C: 2 3 D: 2 2</p> <p><input checked="" type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D Clear Answer Mark For Review</p>	<p>Explanation: *x and *y are used to access or modify the values stored at the addresses x and y point to also called Dereferencing Pointer. y=x means y now points to the same location as x which is 'a' *y=3 changes the value of location y is pointing(x and a) to 3 &a and &b is the address of their corresponding value which is 1 & 2. but when g will run a new value is 3 and y changes the a not b value so y is still 2, hence... 3 2</p>
<p>Q. No. 3. What is the output of the following C program?</p> <pre>#include<stdio.h> void g(int *x, int *y) { *y=x; *y = 3; } int a = 1, b = 2; int main() { g(&a, &b); printf("%d %d\n", a, b); return 0; }</pre> <p>A: 3 2 B: 3 1 C: 2 3 D: 2 2</p> <p><input checked="" type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D Clear Answer Mark For Review</p>	<p>Explanation: *x and *y are used to access or modify the values stored at the addresses x and y point to also called Dereferencing Pointer. y=x means y now points to the same location as x which is 'a' *y=3 changes the value of location y is pointing(x and a) to 3 &a and &b is the address of their corresponding value which is 1 & 2. but when g will run a new value is 3 and y changes the a not b value so y is still 2, hence... 3 2</p>
<p>Q. No. 4. What is the output of the following program?</p> <pre>#include <stdio.h> int main() { int x; if(x=1) printf(" Good "); else printf(" Bad"); }</pre>	

	<p>Q. No. 4. What is the output of the following program?</p> <pre>#include <stdio.h> int main() { int x; if(x=1) printf(" Good "); else printf(" Bad"); return(0); }</pre> <p>A: Unpredictable result as x is not initiated B: Always prints Good C: Compile time error D: Always prints Bad</p> <p><input checked="" type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D Clear Answer Mark For Review</p>	
	<p>Q. No. 5. What is the output of the following C program?</p> <pre>#include <stdio.h> #define a 10 int main() { printf("%d",a+=2); }</pre> <p>A: 10 B: 12 C: Compile time error D: Runtime error</p>	<p>Explanation: as a is a macro not a variable and macro work is to replace all value of a with 10 so in printf it becomes 10=10+2 which gives compile time error.</p>
<p align="center">Examination Instruction Download Response Sheet</p>		
	<p>Q. No. 5. What is the output of the following C program?</p> <pre>#include <stdio.h> #define a 10 int main() { printf("%d",a+=2); }</pre> <p>A: 10 B: 12 C: Compile time error D: Runtime error</p> <p><input type="radio"/> A <input checked="" type="radio"/> B <input type="radio"/> C <input type="radio"/> D Clear Answer Mark For Review</p>	
	<p>Q. No. 6. What is the output of the following C program?</p> <pre>#include <stdio.h> #define x 2+3 #define y 1+2 int main() { printf("%d",x*y); }</pre> <p>A: 15 <input checked="" type="radio"/> B: 7 C: 8</p>	
	<p>Q. No. 6. What is the output of the following C program?</p> <pre>#include <stdio.h> #define x 2+3 #define y 1+2 int main() { printf("%d",x*y); }</pre> <p>A: 15 B: 7 C: 8 D: Compile time error</p> <p><input checked="" type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D Clear Answer Mark For Review</p>	<p>Explanation: #define is used for macro and its work is to replace very appearance with its macro declaration so its 2+3*1+2 -> 2+3+2 = 7, not (2+3)*(1+2) = 15 (incorrect)</p>
	<p>Q. No. 7. Consider the following C program snippet:</p> <pre>float data; extern float edata; Which one of the following is correct?</pre> <p>A: Both the above statements declare variables B: Both the above statements define variables C: First statement declares data and second statement defines edata D: First statement defines data and second statement declares edata</p> <p><input type="radio"/> A <input type="radio"/> B <input checked="" type="radio"/> C <input type="radio"/> D Clear Answer Mark For Review</p>	

<input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="button" value="Clear Answer"/> <input type="button" value="Mark For Review"/>	
<p>Q. No. 7. Consider the following C program snippet:</p> <pre>float data; extern float edata; Which one of the following is correct?</pre> <p>A: Both the above statements declare variables B: Both the above statements define variables C: First statement declares data and second statement defines edata D: First statement defines data and second statement declares edata</p>	<p>Explanation: float data declares the variable and define it for the program with some space allocated to it. extern float edata declares the name of possible variable named edata but does not define it or allocates space for it. extern tell the program that the edata may be coming from some other files.</p>
<input type="radio"/> A <input type="radio"/> B <input checked="" type="radio"/> C <input type="radio"/> D <input type="button" value="Clear Answer"/> <input type="button" value="Mark For Review"/>	
<p>Q. No. 8. What is the output of the following C code snippet?</p> <pre>int x=1,y=12; if(x ++y); printf("%d",y);</pre> <p>A: 13 B: 1 C: 12 D: Compile time error</p>	<p>Explanation: if evaluation end when x = 1 is true and there is a OR operator so 2nd condition does not matter. hence, y remains 12</p>
<input type="radio"/> A <input type="radio"/> B <input checked="" type="radio"/> C <input type="radio"/> D <input type="button" value="Clear Answer"/> <input type="button" value="Mark For Review"/>	
<p>Q. No. 9. Nested function call activation details are maintained through</p>	
<input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="button" value="Clear Answer"/> <input type="button" value="Mark For Review"/>	
<p>Q. No. 9. Nested function call activation details are maintained through</p> <p>A: Queue B: Stack C: Tree D: Graph</p>	<p>Explanation: Nested function call activation details are maintained through a data structure that allows for last-in, first-out (LIFO) order</p>
<input checked="" type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="button" value="Clear Answer"/> <input type="button" value="Mark For Review"/>	
<p>Q. No. 10. What is the output of the following C code snippet?</p> <pre>char *ptr; char str[]="World"; ptr=str; ptr += 3; printf("%s",ptr);</pre> <p>A: rld B: ld C: Wor D: World</p>	<p>Explanation: ptr initially points to the 'W'. After incrementing ptr by 3, it points to 'l'. So, when we print the string from ptr, it starts from 'l' and prints "rld" until the end of the string. NOTE-> WHILE PRINTING SUBSTRING IN C. STARTING POINTING MEANS, (STARTING POINT -1).</p>
<input checked="" type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="button" value="Clear Answer"/> <input type="button" value="Mark For Review"/>	
<p>Q. No. 11. What is the output of the following C code snippet?</p> <pre>int x[2][3]={{1},{2,1,0}}; printf("%d\n",x[1][0]);</pre>	
<input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="button" value="Clear Answer"/> <input type="button" value="Mark For Review"/>	
<p>Q. No. 11. What is the output of the following C code snippet?</p> <pre>int x[2][3]={{1},{2,1,0}}; printf("%d\n",x[1][0]);</pre> <p>A: 0 B: 2 C: 1 D: Garbage value</p>	<p>Explanation: x is a 2D array with 2 rows and 3 columns. The array is initialized with two sets of values: The first set {1} initializes the first row, resulting in x[0] being {1, 0, 0} (remaining values are set to 0 by default). The second set {2, 1, 0} initializes the second row, resulting in x[1] being {2, 1, 0}</p>
<input checked="" type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="button" value="Clear Answer"/> <input type="button" value="Mark For Review"/>	
<p>Q. No. 12. What is the output of the following C code snippet?</p> <pre>int a; a='z'-'w'; printf("%d\n",a);</pre> <p>A: Compilation error B: 3 C: Garbage Value D: 4</p>	<p>Explanation: The ASCII value of 'z' is 122. The ASCII value of 'w' is 119. Subtracting these values: 122 - 119 = 3.</p>
<input checked="" type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="button" value="Clear Answer"/> <input type="button" value="Mark For Review"/>	
<p>Q. No. 13. In C language, break statement cannot be used with</p> <p>A: for B: while C: if</p>	

Q. No. 13. In C language, break statement cannot be used with	
A: for B: while C: if D: switch	
Q. No. 14. What is the output of the following C program snippet?	
<pre>int i=0x10+010+20; printf("%d\n",j);</pre> A: 40 B: 22 C: 44 D: Compile time error	
Q. No. 15. What is the output of the following C code snippet?	
<pre>#include <stdio.h> int main() { int x=0,y=1; x=x*y; y=y*x; printf("%d %d",x,y); return(0); }</pre> A: 0 1 B: 1 0 C: 1 1 D: 0 0	Explanation: In C, the ^ operator is a bitwise XOR (exclusive OR) operator, not a power operator. C does not have a built-in operator for exponentiation (raising to a power). XOR (^) is a bitwise operator that returns 1 when the bits are different and 0 when they are the same. 0 ^ 1 in binary is 0000 ^ 0001, which results in 0001 (or 1 in decimal). So, x becomes 1. Now, x is 1 and y is 1. 1 ^ 1 in binary is 0001 ^ 0001, which results in 0000 (or 0 in decimal). So, y becomes 0.
Q. No. 16. Which of the following is not a function of stack?	
A: Function call B: Infix to postfix conversion C: Balancing symbols D: Searching	Explanation: Stacks are not designed for efficient searching operations; other data structures like arrays or hash tables are better suited for that purpose.
Q. No. 17. Inorder traversal of _____ leads to sorted list of elements as output	
A: Binary tree B: Binary search tree C: Heaps D: Full binary tree	Explanation: In an inorder traversal of a binary search tree, the elements are visited in ascending order, resulting in a sorted list.
Q. No. 18. Inserting and deleting an element into the queue is termed as _____ and _____ respectively	
A: Dequeue, Enqueue B: Enqueue, Dequeue C: Enqueue, Overflow D: Overflow, underflow	Explanation: Inserting an element into the queue is called Enqueue. Deleting an element from the queue is called Dequeue.
Q. No. 19. _____ is not a divide and conquer algorithm	
A: Merge sort B: Quick sort C: Heap sort D: Binary search	Explanation: Unlike the other three options (Merge sort, Quick sort, and Heap sort), which are divide and conquer algorithms, Binary search is not. Binary search is a search algorithm that efficiently finds a target value within a sorted array by repeatedly dividing the search space in half.

100202020003097 CCA1-201/ 000001 10.9020

Session 0

Q. No.20. What data structure is used for breadth first traversal of a graph?	<p>A: queue B: stack C: list D: none of the above</p> <p><input checked="" type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D Clear Answer Mark For Review</p>	<p>Explanation: When performing breadth-first traversal (also known as level-order traversal) of a graph, we use a queue data structure to keep track of the nodes to visit next. The queue ensures that we process nodes in the order they were encountered at each level of the graph.</p>
Q. No.21. Height balanced binary search tree is _____	<p>A: AVL tree B: Red-black tree C: Lemna tree D: Binary tree</p> <p><input checked="" type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D Clear Answer Mark For Review</p>	<p>Explanation: An AVL tree (Adelson-Velsky and Landis tree) is a self-balancing binary search tree where the height difference between the left and right subtrees of any node is at most 1.</p>
Q. No.22. Binding of data members and member functions into a single unit is called as _____	<p>A: Inheritance B: Polymorphism C: Encapsulation D: Genericity</p> <p><input type="radio"/> A <input checked="" type="radio"/> B <input type="radio"/> C <input type="radio"/> D Clear Answer Mark For Review</p>	<p>Explanation: Encapsulation refers to the bundling of data (attributes) and methods (functions) that operate on that data into a single unit (usually a class or object).</p>
<p>Q. No.23. Keywords are _____ of the programming language</p> <p>Examination Instruction Download Response Sheet</p>		
Q. No.22. Binding of data members and member functions into a single unit is called as _____	<p>A: Inheritance B: Polymorphism C: Encapsulation D: Genericity</p> <p><input type="radio"/> A <input checked="" type="radio"/> B <input type="radio"/> C <input type="radio"/> D Clear Answer Mark For Review</p>	
Q. No.23. Keywords are _____ of the programming language	<p>A: Constants B: Identifiers C: Reserved words D: Literals</p> <p><input type="radio"/> A <input checked="" type="radio"/> B <input type="radio"/> C <input type="radio"/> D Clear Answer Mark For Review</p>	<p>Explanation: reserved words (also known as keywords) have predefined meanings and cannot be used for other purposes, such as naming variables or functions. They form the basic building blocks of a program's syntax.</p>
Q. No.24. Members of C++ class are by default	<p>A: private B: public C: protected D: shared</p>	
Q. No.24. Members of C++ class are by default	<p>A: private B: public C: protected D: shared</p> <p><input checked="" type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D Clear Answer UnMark</p>	<p>Explanation: In C++, class members (such as variables and functions) have different access levels. By default: Public members are accessible from outside the class. Private members are only accessible within the class itself. Protected members are accessible within the class and its derived classes</p>
Q. No.25. If Triangle class is derived from Shape class, which one of the following is appropriate way of defining constructor in Triangle class	<p>A: Triangle(int a,int b):Shape(a) { } B: Shape(int a,int b):Triangle(a) { } C: Triangle(int a), Shape(int b) { } D: Shape(int a), Triangle(int b) { }</p> <p><input checked="" type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D Clear Answer Mark For Review</p>	<p>Explanation: This means that the Triangle constructor takes two parameters, a and b. It initializes the Shape part of Triangle using Shape(a). This is the correct syntax for initializing a base class constructor.</p>
Q. No.26. Which one of the following operator cannot be overloaded in C++?	<p>A: * B: .* C: >> D: -></p>	

	<p>Q. No.26. Which one of the following operator cannot be overloaded in C++?</p> <p>A: *</p> <p>B: .*</p> <p>C: >></p> <p>D: -></p>	<p>Explanation: The * operator cannot be overloaded. This operator is used to access members of an object through a pointer-to-member and its behavior is tightly integrated with the structure of the language.</p>
<p><input type="radio"/> A <input checked="" type="radio"/> B <input type="radio"/> C <input type="radio"/> D Clear Answer Mark For Review</p>		
	<p>Q. No.27. Create a class titled Triangle with private non-static data fields named base and height. The Triangle class contains a public non-static function named displayArea() whose header is void Triangle::displayArea(). This function calculates area of triangle and displays the same. Which one of the following correctly invokes this member function over Triangle object?</p> <p>A: Triangle *obj=displayArea();</p> <p>B: Triangle tobj=displayArea();</p> <p>C: Triangle tobj, *tptr=&tobj; tptr->displayArea();</p> <p>D: Triangle *tptr; tptr.displayArea();</p>	<p>Explanation: This option correctly creates a Triangle object (tobj) and a pointer (tptr) that points to tobj. It then correctly invokes displayArea() using the pointer. This is a valid way to call the member function.</p>
<p><input type="radio"/> A <input type="radio"/> B <input checked="" type="radio"/> C <input type="radio"/> D Clear Answer Mark For Review</p>		
	<p>Q. No.28. Which one of the following precisely defines an exception?</p> <p>A: Run time error</p> <p>B: Compile time error</p>	
	<p>Q. No.28. Which one of the following precisely defines an exception?</p> <p>A: Run time error</p> <p>B: Compile time error</p> <p>C: Memory error</p> <p>D: I/O error</p>	<p>Explanation: An I/O error occurs during input/output operations, such as reading from or writing to files, sockets, or streams. Examples include file not found, permission denied, or disk full errors. I/O errors can lead to exceptions (e.g., IOException), and this option precisely defines an exception related to input/output operations.</p>
<p><input type="radio"/> A <input type="radio"/> B <input checked="" type="radio"/> C <input type="radio"/> D Clear Answer Mark For Review</p>		
	<p>Q. No.29. Inline functions are preferred when</p> <p>A: Function is small and want to avoid function call overhead</p> <p>B: Function is complex with many nested loops</p> <p>C: Function has many static variables</p> <p>D: Function is recursive</p>	
<p><input checked="" type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D Clear Answer Mark For Review</p>		
	<p>Q. No.30. What is the output of the following C++ code?</p> <pre>#include<iostream> using namespace std; class PC { public: void print() { cout <<" Inside PC"; } }; class QC : public PC { }</pre>	
	<p>Q. No.30. What is the output of the following C++ code?</p> <pre>#include<iostream> using namespace std; class PC { public: void print() { cout <<" Inside PC"; } }; class QC : public PC { public: void print() { cout <<" Inside QC"; } }; class RC : public QC { }; int main(void) { RC robj; robj.print(); return 0; }</pre> <p>A: Inside PC</p> <p>B: Inside QC</p> <p>C: Compile time error</p> <p>D: Inside PC Inside QC</p>	
<p><input checked="" type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D Clear Answer Mark For Review</p>		
	<p>Q. No.31. _____ is derived by using Insert_end() and Delete_first() functions in a single linked list</p>	

<input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="button" value="Clear Answer"/> <input type="button" value="Mark For Review"/>	
<p>Q. No.31. _____ is derived by using Insert_end() and Delete_first() functions in a single linked list</p> <p>A: Stack B: Queue C: Dqueue D: Tree</p>	<p>Explanation: In a queue, elements are added at the end (enqueue operation) and removed from the front (dequeue operation). Insert_end() is equivalent to the enqueue operation, delete_first() is equivalent to the dequeue operation.</p>
<input checked="" type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="button" value="Clear Answer"/> <input type="button" value="Mark For Review"/>	
<p>Q. No.32. _____ protocol finds the MAC address of a host from its known IP address.</p> <p>A: ARP B: RARP C: ICMP D: IGMP</p>	<p>Explanation: (Address Resolution Protocol): ARP is used to find the MAC address of a host when its IP address is known. It resolves IP addresses to MAC addresses within a local network.</p>
<input checked="" type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="button" value="Clear Answer"/> <input type="button" value="Mark For Review"/>	
<p>Q. No.33. The multiple access method used in GSM cellular technology</p> <p>A: FDMA & CDMA B: CDMA & TDMA C: FDMA & TDMA D: IGMP</p>	<p>Explanation: GSM technology primarily uses a combination of: FDMA: To divide the frequency bands into channels. TDMA: To allocate different time slots to different users on the same frequency band.</p>
<input type="radio"/> A <input type="radio"/> B <input checked="" type="radio"/> C <input type="radio"/> D <input type="button" value="Clear Answer"/> <input type="button" value="Mark For Review"/>	
<p>Q. No.34. In a data communications system, the information to be communicated is the _____.</p> <p>A: Medium B: Protocol C: Message D: Transmission</p>	<p>Explanation: This refers to the actual information or data that is being communicated between the sender and receiver. It is the content of the communication.</p>
<input type="radio"/> A <input type="radio"/> B <input checked="" type="radio"/> C <input type="radio"/> D <input type="button" value="Clear Answer"/> <input type="button" value="Mark For Review"/>	
<p>Q. No.35. If the least significant bit of the first byte is 1, the Ethernet address is _____.</p> <p>A: multicast</p>	<p>Explanation: A) Multicast: Correct. If the LSB of the first byte is 1, it indicates that the Ethernet address is a multicast address. B) Broadcast: Incorrect. A broadcast address in Ethernet is a specific type of multicast address where all bits are set to 1 (FF:FF:FF:FF:FF:FF). C) Unicast: Incorrect. A unicast address is indicated by an LSB of 0 in the first byte. D) Geocast: Incorrect. Geocast is not a standard type of Ethernet address.</p>
<input checked="" type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="button" value="Clear Answer"/> <input type="button" value="Mark For Review"/>	
<p>Q. No.36. _____ is the combination of an IP address and a port number in networking.</p> <p>A: transport address B: network address</p>	<p>Explanation: In networking, a combination of an IP address and a port number uniquely identifies a specific process or service running on a device within a network. This combination is known as a socket. A socket allows for the communication between different processes, either within the same machine or over a network. It is essential for establishing a connection and ensuring that data is sent to the correct application.</p>

<p>Q. No.37. The error detection method which uses one's complement arithmetic is _____.</p> <p>A: Checksum B: CRC C: Simple parity check D: Two-dimensional parity check</p> <p><input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D Clear Answer UnMark</p>	<p>Explanation: In networking and data storage, a checksum is a small-sized datum derived from a block of digital data for the purpose of detecting errors that may have been introduced during its transmission or storage. Checksum calculation often uses one's complement arithmetic.</p>
<p>Q. No.38. The inter frame space, contention window, and acknowledgments are used in which access method to avoid collisions</p> <p>A: CSMA/CD B: FDMA C: CSMA/CA D: TDMA</p> <p><input type="radio"/> A <input type="radio"/> B <input checked="" type="radio"/> C <input type="radio"/> D Clear Answer Mark For Review</p>	<p>Explanation: CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance): This method is used to avoid collisions by using the following mechanisms: Inter-frame space (IFS): A time interval used to space out transmissions. Contention window: A range of time slots from which a station randomly chooses to wait before attempting to transmit. Acknowledgments: Sent by the receiver to confirm the successful reception of data.</p>
<p>Q. No.39. How many bits is the physical address used by Ethernet?</p> <p>A: 32-bit B: 48-bit C: 64-bit D: 128-bit</p> <p><input type="radio"/> A <input checked="" type="radio"/> B <input type="radio"/> C <input type="radio"/> D Clear Answer Mark For Review</p>	<p>Explanation: Ethernet devices use a 48-bit MAC (Media Access Control) address to uniquely identify each device on a network. This address is also known as a hardware address or physical address. MAC Address Format: A MAC address is typically represented as six groups of two hexadecimal digits, separated by colons or hyphens (e.g., 00:1A:2B:3C:4D:5E or 00-1A-2B-3C-4D-5E). Therefore, the correct answer is B, 48-bit.</p>
<p>Q. No.40. The headers are _____, when the data packet is forwarded from the upper to the lower layers.</p> <p>A: Rearranged B: Removed C: Added D: Modified</p> <p><input type="radio"/> A <input type="radio"/> B <input checked="" type="radio"/> C <input type="radio"/> D Clear Answer Mark For Review</p>	<p>Explanation: When a data packet is forwarded from the upper to the lower layers in the OSI model or TCP/IP model, each layer adds its own header to the packet. This process is known as encapsulation. Application Layer: Data is generated and encapsulated with an application layer header. Transport Layer: The data is encapsulated with a transport layer header (e.g., TCP or UDP header). Network Layer: The transport layer data is encapsulated with a network layer header (e.g., IP header). Data Link Layer: The network layer data is encapsulated with a data link layer header (e.g., Ethernet header). Each layer adds its own specific header, which helps in the delivery and processing of the packet as it moves through the network stack.</p>
<p>Q. No.41. A central controller or hub is required in which type of topology?</p> <p>A: Mesh B: Bus C: Star D: Ring</p> <p><input type="radio"/> A <input type="radio"/> B <input checked="" type="radio"/> C <input type="radio"/> D Clear Answer Mark For Review</p>	<p>Explanation: In a star topology, all devices are connected to a central device, which is typically a hub or a switch. This central controller manages and directs the data flow.</p>
<p>Q. No.42. Process is _____</p> <p>A: program in High level language kept on disk B: contents of main memory C: a program in execution D: a job in secondary memory</p> <p><input type="radio"/> A <input type="radio"/> B <input checked="" type="radio"/> C <input type="radio"/> D Clear Answer Mark For Review</p>	<p>Explanation: A: A program in High-level language kept on disk: This describes a program that is not currently being executed but is stored on a disk. This is not a process but a program or source code. B: Contents of main memory: While processes reside in main memory when they are executed, not all contents of main memory are processes. This description is too broad. C: A program in execution: This accurately describes a process. When a program is loaded into memory and begins execution, it becomes a process. D: A job in secondary memory: Secondary memory typically refers to storage devices like hard drives or SSDs. Jobs in secondary memory are not currently being executed, hence they are not processes.</p>
<p>Q. No.43. Which of the following describes the ability of an OS to support multiple, concurrent paths of execution within a single process?</p> <p>A: Multithreading B: Multiprocessing</p> <p><input type="radio"/> A <input type="radio"/> B <input checked="" type="radio"/> C <input type="radio"/> D Clear Answer Mark For Review</p>	

Q. No. 43. Which of the following describes the ability of an OS to support multiple, concurrent paths of execution within a single process?

A: Multithreading
B: Multiprocessing
C: Multitasking
D: Multiprogramming

Explanation:
A: Multithreading: Multithreading refers to the ability of a CPU, or a single process, to manage multiple threads of execution concurrently. Each thread shares the same process resources but can execute independently.
B: Multiprocessing: Multiprocessing refers to the use of two or more CPUs within a single computer system. It involves multiple processors handling multiple processes simultaneously.
C: Multitasking: Multitasking refers to the ability of an OS to execute multiple tasks (processes) simultaneously. This can be achieved through context switching where the CPU switches between different tasks quickly.
D: Multiprogramming: Multiprogramming is a method where multiple programs are loaded into memory and the OS manages their execution to ensure efficient use of resources. It does not inherently imply concurrency within a single process.

Q. No. 44. What is not shared by threads?

A: Code
B: Data
C: Files
D: Registers

Explanation:
A) Code: Threads share the same code segment since they are part of the same process.
B) Data: Threads share the data segment, including global variables and the heap.
C) Files: Threads share open file descriptors, meaning they can access and manipulate the same files.
D) Registers: Registers are not shared by threads. Each thread has its own set of registers, including the program counter, stack pointer, and other CPU-specific registers.

Q. No. 45. High page faults leads to --

A: Swapping
B: Compaction
C: Thrashing
D: External Fragmentation

Explanation:
A) Swapping: Swapping is the process of moving processes between main memory and a storage device. While swapping can occur due to page faults, high page faults specifically lead to a more severe condition known as thrashing.
B) Compaction: Compaction is the process of rearranging memory contents to eliminate fragmentation. This is related to memory management but not directly caused by high page faults.
C) Thrashing: Thrashing occurs when a system spends more time handling page faults and swapping pages in and out of memory than executing actual processes. High page faults can cause the system to constantly swap pages, leading to thrashing and severely degraded performance.
D) External Fragmentation: External fragmentation occurs when free memory is divided into small, noncontiguous blocks. This is related to memory allocation but not directly to high page faults.

Q. No. 46. What is compaction?

A: A technique for overcoming internal fragmentation
B: A paging technique
C: A technique for overcoming external fragmentation
D: A technique for overcoming fatal error

<p>B: Compaction C: Thrashing D: External Fragmentation</p> <p><input type="radio"/> A <input checked="" type="radio"/> B <input type="radio"/> C <input type="radio"/> D Clear Answer Mark For Review</p>	
<p>Q. No.46. What is compaction?</p> <p>A: A technique for overcoming internal fragmentation B: A paging technique C: A technique for overcoming external fragmentation D: A technique for overcoming fatal error</p> <p><input type="radio"/> A <input checked="" type="radio"/> B <input type="radio"/> C <input type="radio"/> D Clear Answer Mark For Review</p>	<p>Compaction is a technique used in memory management to overcome external fragmentation. External fragmentation occurs when free memory is divided into small, noncontiguous blocks, making it difficult to allocate large contiguous blocks of memory to processes.</p> <p>A) A technique for overcoming internal fragmentation: Internal fragmentation happens when allocated memory may contain unused space. Compaction does not address this issue.</p> <p>B) A paging technique: Paging is a memory management scheme that eliminates the need for contiguous allocation of physical memory. Compaction is not related to paging.</p> <p>C) A technique for overcoming external fragmentation: Compaction rearranges memory contents to consolidate free memory into a single contiguous block, thus overcoming external fragmentation.</p> <p>D) A technique for overcoming fatal error: Fatal errors are critical system errors that usually require a restart. Compaction is not a technique to handle fatal errors.</p>
<p>Q. No.47. short term scheduler is also known as _____</p> <p>A: cpu scheduler B: job scheduler C: middle term scheduler D: none of these</p> <p><input checked="" type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D Clear Answer Mark For Review</p>	<p>Explanation: The short-term scheduler is also known as the CPU scheduler. Its primary responsibility is to select which of the ready, in-memory processes will be executed next by the CPU.</p> <p>A) CPU scheduler: Correct. The short-term scheduler decides which process will run on the CPU next. It operates frequently, making decisions at a high rate to ensure efficient CPU utilization.</p> <p>B) Job scheduler: Incorrect. The job scheduler is typically another name for the long-term scheduler, which decides which jobs or processes are admitted to the system for processing.</p> <p>C) Middle-term scheduler: Incorrect. This typically refers to the medium-term scheduler, which is responsible for swapping processes in and out of the main memory to manage the degree of multiprogramming.</p>
<p>Q. No.48. Find the wrong statement about multilevel queue scheduling</p> <p>A: Ready queue is partitioned into separate queues</p> <p><input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D Clear Answer Mark For Review</p>	<p>Explanation: Multilevel Queue Scheduling organizes processes into separate queues based on priority or other criteria, allowing for efficient management of diverse workloads.</p>
<p>D: Scheduling must be done between the queues</p> <p><input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D Clear Answer Mark For Review</p>	
<p>Q. No.49. Accessing speed is higher for _____</p> <p>A: Solid-state disks B: Main memory C: Cache D: Registers</p> <p><input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input checked="" type="radio"/> D Clear Answer Mark For Review</p>	<p>Explanation: Accessing speed varies significantly among different types of memory and storage within a computer system. The correct ranking from fastest to slowest is typically as follows: registers, cache, main memory, and then solid-state disks (SSDs).</p> <p>Registers are the fastest form of memory. They are located within the CPU and are accessed directly by the processor with virtually no delay.</p>
<p>Q. No.50. Virtual memory is</p> <p>A: extremely large main memory B: extremely large secondary memory C: illusion of extremely large memory D: a type of memory used in super computers</p> <p><input type="radio"/> A <input type="radio"/> B <input checked="" type="radio"/> C <input type="radio"/> D Clear Answer Mark For Review</p>	<p>Explanation: Virtual memory is a memory management technique that creates an illusion of an extremely large memory space by using a combination of hardware and software. This allows a computer to compensate for physical memory shortages, by temporarily transferring data from random access memory (RAM) to disk storage.</p> <p>Correct. Virtual memory creates an illusion for the user of a very large main memory by using disk storage to extend the available memory beyond the physical limits of the RAM.</p>