C++ and Data Structures (60 Minutes)

1. Evaluate the following for fn (7);

```
int fn(int v)
    {
    if(v==1 || v==0)
    return 1;
    if(v%2==0)
    return fn(v/2)+2;
    else
    return fn(v-1)+3;
    }
```

   1. 11
   2. 10
   3. 9
   4. 1

2. My salary was increased by 15%!"
   Which of the following statement will EXACTLY print the line of text above?
   1. printf("My salary was increased by 15%!\n");
   2. printf("My salary was increased by 15'%'!\n");
   3. printf("\"My salary was increased by 15/%\!\"\n");
   4. printf("\"My salary was increased by 15%%!\"\n");
3. Why should we not use non-integer real numbers as counters in loop?
   1. Because they are not allowed in loop
   2. A slight imprecision in repeating numbers can cause the loop to repeat infinitely.
   3. Because they use more space
   4. Because they are slow and less efficient than integers
4. If a process does *x.signal*, where *x* is a condition variable of a monitor and no process is awaiting condition x, What will happen?
   1. The signal operation has no effect
   2. The next process do x.wait does not get blocked
   3. The next process do x.wait get blocked
   4. None of the above
5. Is the below statement is True/False?
   The sequence of instructions (used in Processes) using mutex semaphore

```
Up(mutex);
        Critical section
Down(mutex);
```

   satisfies mutual exclusion.
   1. True
   2. False
   3. Cannot say
   4. None of the above
6. What will be the output of the following program?

```
#include<stdio.h>
#include<string.h>
char *getptr()
{
 static char ptr[10] = "123456789";
 return ptr;
}
void main()
{
 char *ptr="00000";
 strcpy(getptr()+4,ptr);
 ptr=getptr();
 strcpy(ptr,"12345");
 printf("%s",getptr());
}
```

   1. 123456789
   2. 1234500000
   3. 12345
   4. Run time error
7. What will be the output of the following program?

```
#include<stdio.h>
void main()
{
  int x = 5;
  int y = 2;
  char op = '*';
  switch (op)
  {
    default : x += 1;
    case '+' : x += y;
    case '-' : x -= y;
  }
  printf("%d",x);
}
```

   1. Compile time error
   2. Run time error
   3. 5
   4. 6
8. What will be the output of the following Code?

```
int pro=0, count=0;
while (pro<2500)
{
  pro*=5;
      count++;
}
```

   1. Syntax error: while statement is not valid
   2. The operator *= does not exist
   3. The count variable is initialized incorrectly
   4. It has an infinite loop
9. What will be the output of following program?

```
#include<stdio.h>
#define INDX(x) (++indx[x][ptr[x-1]],&indx[x][ptr[x-
1]])
void main()
{
  int indx[6]={0,1,2,3,4,5};
  int i=10;
  char arr[][20]={"%hs not %s %s","%dbool",
  "?n%shis? "};
  char *ptr[3];
  ptr[0]=&arr[0][0];
  ptr[1]=&arr[1][0];
  ptr[2]=&arr[2][0];
   printf(INDX(1),INDX(3),INDX(2));
}
```

   1. this is bool
   2. is not this cool
   3. is not this? cool
   4. Compile time error

10. What will be the output of following program?

```
#include<stdio.h>
void main()
{
  struct num
  {
    int x,y;
  } val[4] = {1,1,2,3,4,5,6,7};
  struct num *ptr = val;
  int i=0;
  for(;i<4;i++) {
      ptr->x = ptr->y, ++ptr++->y;
      printf("%d,%d ", val[i].x, val[i].y);
  }
}
```

1. 1 1 2 3 4 5 6 7
2. 1 2 3 4 5 6 7 8
3. 1 2 3 4 5 6 7
4. 1 1 2 3 4 5 6

11. Which of the following is/are the special functions a C++ compiler can create implicitly?
    1. The default constructor
    2. The copy constructor and the destructor
    3. The operator=() function
    4. All of the above
12. What does the code do?

```
strcat(an_array, "This");
```

    1. Copies "This" into an_array
    2. Adds "This" to the end of an_array
    3. Compares an_array and "This"
    4. Both 1 and 2
13. What will be the output of the following program?

```
#include<iostream.h>
int count=0;
class object
{
  public :
  object(){count++;}
  ~object(){count--;}
};
int main()
{
  object A,B,C,D,E;
  object F,G;
{
        object H;
}
  cout<<count;
  return 0;
}
```

    1. 0
    2. 1
    3. 7
    4. 6
14. Consider A and B as two operands, and " +" as the operator, the presentation AB+ is called _____.
    1. Infix
    2. Suffix
    3. Prefix
    4. Postfix
15. In case of a copy constructor, which of the following is true?
    a. Used to instantiate an object from another existing object
    b. To copy one object to another existing object
    1. Only a
    2. Only b
    3. Both a and b
    4. None of the above
16. What will be happens if you use the delete keyword on a null pointer?
    1. The apocalypse
    2. A crash may occur
    3. Undefined behavior
    4. Nothing happens
17. Which of the following statement is true?
    1. Overridden functions are in different scopes; whereas overloaded functions are in same scope.
    2. Overriding is needed when derived class function has to do some added or different job than the base class function.
    3. Overloading is used to have same name functions, which behave differently depending upon parameters passed to them.
    4. All of the above
18. What will be the output of following program?

```
#include<iostream.h>
namespace NS1
{
  int f(int n) {return n*4;}
}
namespace NS2
{
  int f(double n) {return n*7;}
}
void main()
{
  using NS1::f;
  int a=f(1.0);
  using NS2::f;
  int b=f(1.0);
  cout<<a<<b;
}
```

    1. 74
    2. 77
    3. 44
    4. 47
19. What will be the output of the following program?

```
class Window
{
    public: virtual void Create() { cout <<"Base
    class Window"; }
};
class CommandButton : public Window
{
    public: void Create() { cout<<"Derived class
    Command Button"; }
 };
void main()
{
    Window *x, *y;
    x = new Window();
    x->Create();
    y = new CommandButton();
    y->Create();
}
```

    1. Base class Window
    2. Derived class Command Button
    3. Base class Window Derived class Command Button

4. Derived class Command Button Base class Window

20. Which of the following is an example where copy constructor is needed?
    a. User-defined copy constructor is required when deep copy is required
    b. Cloning of objects in Prototype pattern
    1. Only a
    2. Only b
    3. Both a and b
    4. None of the above

21. In addition to c-style, which casts can be used to cast an int into an enum?
    1. dynamic_cast
    2. static_cast
    3. reinterpret_cast
    4. None of the above

22. What will be the output of following program?
```
#include<iostream.h>
void main()
{
  int a;
  bool b;
  a = 10 > 20;
  b = 10 >= 20;
  cout<<a<<" "<<b<<endl;
}
```
    1. 0 0
    2. 0 1
    3. 1 1
    4. 1 0

23. Which lines of code below should cause the program to be undefined?
```
 1 struct Foo
 2 {
 3   virtual ~Foo() {}
 4 };
 5
 6 struct Bar : public Foo
 7 {
 8 };
 9
10 int main(int argc, char** argv)
11 {
12   Foo* f = new Bar;
13   delete f;
14   f = 0;
15   delete f;
16
17   Foo* fa = new Bar[10];
18   delete fa;
19   fa = 0;
20   delete fa;
21
22   return 0;
23 }
```
    1. 13
    2. 15
    3. 18
    4. 20

24. What will be the output of following program?
```
#include<iostream.h>
void main()
{
  int arr[][3]={0,10,20,30,40,50};
  int *a = &arr[0][0];
  cout<<arr[1][2]<<" "<<*(a+3);
}
```
    1. 30 50
    2. 50 30
    3. Compile time error
    4. Run time error

25. Which of the following statements is NOT true?
    a. Operators can be overloaded when both operands are of built-in types.
    b. Operators can be overloaded when one operand is of a built-in type and the other is of user-defined type.
    c. Operators can be overloaded when both operands are of user-defined types.
    1. Only a
    2. Only b
    3. Both b and c
    4. Both a and c

26. When must template functions have explicit template parameters?
    1. When the template types cannot be inferred
    2. Never, the template types can always be inferred
    3. Always
    4. None of the above

27. What will be the output of the following program?
```
#include<iostream.h>
int main()
{
  int x = 0x1000;
  x = x << 32;
  cout << hex << x ;
  return 0;
}
```
    1. 32
    2. 1000
    3. 0xFFFFFFFF
    4. 0x00000000

28. A friend function can access _____.
    1. Not even public members of class
    2. Only public members of class
    3. Only public and protected members of class
    4. Public, protected and even private members of the class

29. Which one of the following is true regarding the compiling and running of the following line
```
ptr->Fn();
```
    If ptr is a pointer to type Q and Fn() is a virtual function in class Q?
    1. function name and function definition both matched to type of pointer at compile time
    2. function name matched to pointer type at compile time;
       function definition matched to type pointer points to at run time
    3. function name matched to pointer type at run time;
       function definition matched to type pointer points to earlier, at compile time
    4. function name and function definition both matched to type of object pointer points to at run time

30. Consider the following code:
```
void negate(int& x)
 {
    x = -1 * x;
 }
```

What would happen during a call "negate(y);" in main? (y is an integer variable)
1. x is negated, y is not
2. y is negated, x is not
3. Both x and y are negated
4. Compile time error

31. Queues serve a major role in _____.
1. Simulation of arbitrary linked list
2. Simulation of limited resource allocation
3. Simulation of recursion
4. Expression evaluation

32. The postfix form of A-B(C*D$E) is _____.
1. ABCDE$-/*
2. ABCDE/-*$
3. AB/C*DE$
4. ABCDE$*/-

33. Which of the following sorting method is stable?
1. Straight insertion sort
2. Shell sort
3. Heap Sort
4. Binary insertion sort

34. Modern filesystems, like ReiserFS and XFS, use which of the following structure to organize their data for efficient access?
1. B+ tree
2. B- tree
3. Hash table
4. Array

35. Linked list is not suitable for which of the following problems?
1. Insertion sort
2. Binary search
3. Quick sort
4. Polynomial manipulation

36. Which of the following is NOT a good characteristic of a hash function?
1. Uniform distribution
2. Easy to compute
3. Handles various sized key spaces
4. Frequent collisions

37. Which of the following is true about connected graph?
a. It cannot be portioned with out removing an edge
b. It contains at least 3 loops.
1. Only a
2. Only b
3. Both a and b
4. None of the above

38. Which of the following is /are a self-balancing binary search trees?
1. AVL tree
2. Red black trees
3. Both a and b
4. None of the above

39. If you have a sorted, balanced binary tree with 15 elements in it, how many steps, maximum, will it take you to decide whether an element is present in the tree or not?
1. Three
2. Four
3. Fifteen
4. Fourteen

40. The average time required to perform a successful sequential search for an element in an array A(1:n) is given by _____.
1. n+1/2
2. n(n+1)/2
3. Log n
4. n*n

41. Which of the following statement is true?
a. Parenthesis are never needed in prefix or postfix expressions.
b. A postfix expression is merely the reverse of the prefix expression.
1. Only a
2. Only b
3. Both a and b
4. None of the above

42. A machine took 200 sec to sort 200 names, using bubble sort. In 800 sec, it can approximately sort _____.
1. 800 names
2. 400 names
3. 750 names
4. 700 names

43. Which of the following is false?
a. Insertion of an element should be done at the last node in a circular list
b. Deletion of an element should be done at the last node in a circular list
1. Only a
2. Only b
3. Both a and b
4. None of the above

44. A circular list can be used to represent _____.
1. Stack
2. Queue
3. B-tree
4. Both 1 and 2

45. A famous quotation of NIKLAUS WIRTH states Algorithm+ Data Structure=_____.
1. Computer
2. Software
3. Program
4. Array

46. What can be said about the array representation of a circular queue when it contains only one element?
1. front=Rear=Null
2. front =Rear+1
3. front=Rear-1
4. front==Rear

47. Let q be the queue of interger defined as follows:

```
#define MAX-Q 500
struct queue
{
  int item[MAX-Q]
  int front,rear;
}
```

To insert an element in the queue, which of the following operation we use?
1. ++q.item[q.rear]=X;
2. q.item[q.rear]++=X;
3. q.item[++q.rear]=X;
4. None of the above

48. Which of the following statement is false?
    a. Linked list are not superior to STL vectors.
    b. Deleting a node in a linked list is a simple matter of using the delete operator to free the nodes Memory
    1. Only a
    2. Only b
    3. Both a and b
    4. None of the above

49. Which of the following is a definition of a printAll() method for the List class that relies on an overloaded << method for Person to print the contents of the list?
    1.
    ```
    void List::printAll()
    {
    for(ListNode *ptr = tail; ptr; ptr = ptr->next)
        cout << *(ptr->person);
    }
    ```
    2.
    ```
    void List::printAll()
    {
    for(ListNode *ptr = tail; ptr; ptr = ptr->next)
        cout << *(ptr.person);
    }
    ```
    3.
    ```
    void List::printAll()
    {
    for(ListNode ptr = head; ptr; ptr = ptr->next)
        cout << (ptr->person);
    }
    ```
    4.
    ```
    void List::printAll()
    {
    for(ListNode *ptr = head; ptr; ptr = ptr->next)
        cout << *(ptr->person);
    }
    ```

50. Complexity of Kruskal`s algorithm for finding the minimum spanning tree of an undirected graph containing n vertices and m edges if the edges are sorted is _____.
    1. O(m)
    2. O(m+n)
    3. O(n)
    4. None of the above