SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
RAMAPURAM CAMPUS, CHENNAI-89
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

18CSC202J-OBJECT ORIENTED DESIGN AND PROGRAMMING

QUESTION BANK

UNIT-1

1 Marks

1. Which of the following explains Polymorphism? (L1) (Page no 19)
A) intfunc(int, int);
Float func1(float,float);
B) intfunc(int);
Intfunc(int);
C) intfunc(float);
Intnew_func();
D) intfunc();
Intnew_func();
Ans: C

2. Find how many bytes are occupied by the following data types in a 32-bit system. (L1) (Page no 70) Reference-5
A) Type int
B) Type long double
C) Type float
D)Type long
Ans: C

3. Which of the following is a important role of a function? (L1) (Page no 210)
A) give a name to a block of code
B) reduce program size
C) accept arguments and provide a return value
D) help organize a program into conceptual units
Ans:D

4. The Unified Modeling Language is ……(L1) (Page no 27)- Reference-5
A) a program that builds physical models.
B) a way to look at the organization of a program
C) the combination of C++ and FORTRAN
D) helpful in developing software systems.
Ans:D

5. In C++, a function contained within a class is called ……..(L1) (Page no 25)
A) a member function

B) an operator
C) a class function
D) a method
Ans:A

6. What happens if the base and derived class contains definition of a function with same prototype? (L1)(Page no 21)

A) Compiler reports an error on compilation
B) Only base class function will get called irrespective of object
C) Only derived class function will get called irrespective of object
D) Base class object will call base class function and derived class object will call derived class object will call derived class function
Ans:D

7. Which one of the following option is correct about the statement given below? The compiler checks the type of reference in the object and not the type of object(L1) (Page no 29)

A) inheritance
B) Polymorphism
C) Abstraction
D) Encapsulation
Ans:B

8. Which of the following functions are performed by a constructor? **(L1)**(Page no 32)

A) Construct a new class
B) Construct a new object
C) Construct a new function
D) Initialize objects
Ans: D

9. Which of the following is the correct class of the object cout(L1) (Page no 38)

A) iostream
B) istream
C) ostream
D) ifstream
Ans: C

10. In UML, diagrams which captures system static structure and provide foundation for other models is called ..... (L1) (Page no 35)

A) Deployment diagram
B) Class diagram
C) Component diagram
D) Object diagram
Ans: B

11. Find the error produced by compiler when private members are accessed? (L1) (Page no 31)

A) Can't access private message
B) Code unreachable
C) Core dumped
D) Bad code
Ans: A

12. Choose the default access specifier for the class member(L1) (Page no 40)

A) public

B) private
C) protected
D) None of the above
Ans: B

13. Which of the following is CPP style type-casting? (L1) (Page no 42)
A) per=total/ (float)m
B) per=total/float(m)
C) per = (float)total/m
D) None of these
Ans: B

14.What is the output of the following program? (L1) (Page no 37)

```
#include<iostream>
using namespace std;
void main()
{
char s() = "SRM";
      *s = 'R';
cout<<s<<endl;
}
```

A) RRM B) SRM C) SRR D) None of these
Ans: A(Page no 37)

15.What does the following statement mean? (L1) (Page no 42)
int (*fp)(char*)

A) pointer to a pointer
B) pointer to an array of chars
C) pointer to function taking a char* argument and returns an int
D) function taking a char* argument and returning a pointer to int
Ans : C

16.Which of the following concepts of OOPS means exposing only necessary information to client? (L1) (Page no 42)
A) Encapsulation
B) Abstraction
C) Polymorphism
D) Data binding
Ans: B

17.Which of the following is illegal? (L1) (Page no 36)
A) int *ip;
B) string s, *sp = 0;
C) int i; double* dp = &i;
D) int *pi = 0;

Ans: C

18.Which member can never be accessed by inherited classes? (L1) (Page no 45)
A) Private member function

B) Public member function
C) Protected member function
D) All can be accessed
Ans: A

19.Analyze the code and choose the correct Ans? (L4) (Page no 37)

```
int a=100, b=200;
int *p=&a, *q=&b;
p=q;
```

A) b is assigned to a
B) p now points to b
C) a is assigned to b
D) q now points to a
Ans: B

20. Mention the size_t integer type in C++ is? (L2) (Page no 52)
A) Unsigned integer of at least 64 bits
B) Signed integer of at least 16 bits
C) Unsigned integer of at least 16 bits
D) Signed integer of at least 64 bits
Ans: c

4 Marks

21. Abstraction and encapsulation are fundamental principles that underlie the object oriented approach to software development. What can you say about the following two statements? (L2) (Page no 22)
I.Abstraction allows us to focus on what something does without considering the complexities of how it works.
II.Encapsulation allows us to consider complex ideas while ignoring irrelevant detail that would confuse us.

A. Neither I nor II is correct
B. Both I and II are correct
C. Only II is correct
D. Only I is correct

Ans: A

22. Class A

```
{ public :
A(void)
{cout <<  "Howzaht" ; }
~ A(void)
{cout << "whatizit " ; }
};
class B:A
{ public:
 ~ B (void)
{
cout  << "WYSIWYG" ;
```

```
}
cout << "YACC" ;
}
};
```

If the main function has the two statements (L2) (Page no 21)

```
B x ;
cout  << " done" ;
```

the output will be

A. Howzhat WYSIWYG YACC Whatizi t done

B. Howzhat WYSIWYG done YACC Whatizit

C. Yl>.CC Whatizit Howzhat WYSIWYG done

D. none of the above

Ans: B

23. The following program (L2) (Page no 31)

```
class abc;
class def
{
int i1 ;                    // statement 1
protected: int i2;   // statement 2
public: int i3;       //statement 3
friend abc;
} ;
class abc
{
public:
void main(def A)
{
cout << (A.i1=3);
cout <<  (A.i2=4);
cout << (A.i3=5);
}
} ;
void main( )
{
def x1 ;
abc x2 ;
x2.mn(x1);
}
```

A. will compile successfully if statement 1 is removed

B. will compile successfully if statement 2 is removed

C. will compile successfully if statement 3 is removed

D.will run successfully and print 34 5

Ans D

24. What will be the output of the following code? (L2) (Page no 36)

Class A

```
{
        int i;
        public : A(int n)
        {
                i=n; cout&lt;&lt;"inside constructor ";
        }
        ~A()
        {
                cout&lt;&lt;"destroying  "&lt;&lt;i;
        }
        void seti(int n)
        {
                i=n;
        }
        int geti()
        {
                return I;
        }
};
void t(A ob)
{
        cout&lt;&lt;"something ";
}
int main()
{
        A a(1);
        t(a);
        cout&lt;&lt;"this is i in main ";
        cout&lt;&lt;a.geti();
}
```

A)inside constructor something destroying 2this is i in main destroying 1

B)inside constructor something this is i in main destroying 1

C)inside constructor something destroying 2this is i in main

D) something destroying 2this is i in main destroying 1

Ans A

25. Match the following concepts and their best possible descriptions. (L2) (Page no 42)

	Concept		Description
i.	overloading	a.	allows to define a class to have properties of another class
ii.	friend	b.	defining a set of similar functions
iii.	constructor	c.	used in dereferencing
iv.	protected	d.	used to give a non-member function access to the private parts of an object
v.	this	e.	a function which is automatically called when an object is created
vi.	inheritance	f.	allows a derived class to have access to the private parts of the base class
		g.	a pointer to the object associated with the current function
		h.	used to obtain object persistence

.

(A) i-b, ii-d, iii-e, iv-f, v-g, vi-a **(B)** i-c, ii-a, iii-e, iv-d, v-h, vi-f
(C) i-c, ii-f, iii-h, iv-a, v-g, vi-d **(D)** i-b, ii-e, iii-c, iv-f, v-g, vi-s

Ans:A

26. Predict the output of following C++ program(L2) (page no 43)

```
#include<iostream>
using namespace std;
class Empty {};
 int main()
{
  cout << sizeof(Empty);
  return 0;
}
```

A. A non-zero value
B. 0
C. Compiler Error
D. Runtime Error

Ans:A

27. Predict the output following program. (L2) (Page no 43)

```
#include<iostream>
using namespace std;
  class Test
```

```
{
   static int x;
   int *ptr;
   int y;
};
 int main()
{
   Test t;
   cout << sizeof(t) << " ";
   cout << sizeof(Test *);
}
```

A. 12 4
B. 12 12
C.8 4
D. 8 8

Ans:C

28. Output of following program? (L2) (page no 43)

```
#include<iostream>
using namespace std;
 class Point {
public:
   Point() { cout << "Normal Constructor calledn"; }
   Point(const Point &t) { cout << "Copy constructor calledn"; }
};
int main()
{
  Point *t1, *t2;
  t1 = new Point();
  t2 = new Point(*t1);
  Point t3 = *t1;
  Point t4;
  t4 = t3;
  return 0;
}
```

A. Normal Constructor called
Normal Constructor called

Normal Constructor called

Copy Constructor called

Copy Constructor called

Normal Constructor called

Copy Constructor called

B. Normal Constructor called
Copy Constructor called

Copy Constructor called

Normal Constructor called

Copy Constructor called

C.Normal Constructor called
Copy Constructor called

Copy Constructor called

Normal Constructor called

D.Normal Constructor called
Copy Constructor called

Copy Constructor called

Ans:C

29.Analyze the Output. (L4) (Page no 44)

```
#include<iostream>
using namespace std;
  class Test
{
public:
  Test();
};

Test::Test()  {
    cout << " Constructor Called. ";
}

void fun() {
  static Test t1;
}

int main() {
    cout << " Before fun() called. ";
    fun();
    fun();
    cout << " After fun() called. ";
    return 0;
}
```

A. Constructor Called. Before fun() called. After fun() called.
B. Before fun() called. Constructor Called. Constructor Called. After fun() called.

C. Before fun() called. Constructor Called. After fun() called.
D.Constructor Called. Constructor Called. After fun() called.Before fun() called.

Ans:C

30. What is the output of following program? (L2) (page no 43)

```
#include <iostream>
using namespace std;

class Point
{
    int x, y;
public:
   Point(const Point &p) { x = p.x; y = p.y; }
   int getX() { return x; }
   int getY() { return y; }
};

int main()
{
    Point p1;
    Point p2 = p1;
    cout << "x = " << p2.getX() << " y = " << p2.getY();
    return 0;
}
```

a.x = garbage value y = garbage value
b. x = 0 y = 0
c. Compiler Error
d.Runtime

Ans: C

12Marks

31. Consider the following class definitions in a hypothetical Object-Oriented language that supports inheritance and uses dynamic binding. The language should not be assumed to be either Java or C++, though the syntax is similar. (L4) (Page no 62)

```
Class P
{
   void f(int i)
   {print(i);
   }
}
```

Class Q subclass of P

```
{
   void f(int i)
   {
      print(2*i);
   }
}
```

Now consider the following program fragment:

```
P x = new Q();
Q y = new Q();
P z = new Q();
x.f(1); ((P)y).f(1); z.f(1);
```

Here ((P)y) denotes a typecast of y to P. The output produced by executing the above program fragment will be

a.121
b.211
c.212
d.222

Ans: D

32. Which of the following is true about the following program(L4) (Page no 72)

```
#include <iostream>
class Test
{
public:
   int i;
   void get();
};
void Test::get()
{
   std::cout << "Enter the value of i: ";
   std::cin >> i;
}
Test t;  // Global object
int main()
{
   Test t;  // local object
   t.get();
   std::cout << "value of i in local t: "<<t.i<<'\n';
   ::t.get();
   std::cout << "value of i in global t: "<<::t.i<<'\n';
   return 0;
}
```

a.Compiler Error: Cannot have two objects with same class name
b**.**Compiler Error in Line "::t.get();"

c. Compiles and runs fine
d. Runtime Error
Ans:C

33. What will be the out of the following program? (L4) (page no 56)

```
#include<iostream.h>
class BixBase
{
 protected:
 int x, y;
 public:
   BixBase(int xx = 0, int yy = 0)
   {
      x = xx;
      y = yy;
   }
   void Show()
   {
      cout<< x * this->y << endl;
   }
};
class BixDerived
{
   private:
      BixBase objBase;
   public:
   BixDerived(int xx, int yy) : objBase(xx, yy)
   {
      objBase.Show();
   }
   ~BixDerived()
   { }
};
int main()
{
   BixDerived objDev(10, 20);
   return 0;
}
```

A.0 B.100 C.200 D.400

Ans C

34.Analyze the Output. (L4) (Page no 74)

```
 #include <iostream>
using namespace std;
class A
{
	int id;
	static int count;
public:
	A() {
		count++;
		id = count;
		cout << "constructor for id " << id << endl;
	}
	~A() {
		cout << "destructor for id " << id << endl;
	}
};

int A::count = 0;

int main() {
	A a[3];
	return 0;
}
```

A.constructor for id 1
constructor for id 2

constructor for id 3

destructor for id 3

destructor for id 2

destructor for id 1

B.constructor for id 1
constructor for id 2

constructor for id 3

destructor for id 1

destructor for id 2

destructor for id 3

C.Compiler Dependent.

D.constructor for id 1
destructor for id 1

Ans:A

35. Analyze the output. (L4) (Page no 53)

```
Class A
{
 int i;
 public : A(int n)
 {
  i=n; cout&lt;&lt;"inside constructor ";
 }
 ~A()
 {
  cout&lt;&lt;"destroying  "&lt;&lt;i;
 }
 void seti(int n)
 {
  i=n;
 }
 int geti()
 {
  return I;
 }
};
void t(A ob)
{
 cout&lt;&lt;"something ";
}
int main()
{
 A a(1);
 t(a);
 cout&lt;&lt;"this is i in main ";
 cout&lt;&lt;a.geti();
}
```

A. inside constructor something destroying 2this is i in main destroying 1

B. inside constructor something this is i in main destroying 1

C. inside constructor something destroying 2this is i in main

D. something destroying 2this is i in main destroying 1

Ans : A

36.Analyze the Output (L4) (Page no 68)

```
template <typename T>
void test(const T&x)
{
   static int count = 0;
   cout &lt;&lt; "x = " &lt;&lt; x &lt;&lt; " count = " &lt;&lt; count &lt;&lt; endl;
   ++count;
```

```
    return;
}

void main()
{
    test<int> (2);
    test<int>(2);
    test<double>(2.2);
}
```

A.x = 2 count = 0
x = 2.2 count = 0
x = 2.2 count = 0
B.x = 2 count = 0
x = 2 count = 0
x = 2.2 count = 0
C.x = 2 count = 0
x = 2 count = 1
x = 2.2 count = 0
D.x = 2 count = 0
x = 2 count = 1
x = 2.2 count = 2

Ans:C

UNIT –II

1 Mark

1. While overloading binary operators using member function, it requires ___ arguments. [L1]Page no:328

a. Zero
b. One
c. Two
d. Three
Ans: b

2. Which of the followings are true about constructors?[L1]Page no:227

1. A class can have more than one constructor.
2. They can be inherited.
3. Their address can be referred.
4. Constructors cannot be declared in protected section of the class.
5. Constructors cannot return values.

a. Only 1, 2, 4
b. 1,2,4,5
c. 1, 3, 5
d. 1, 4, 5
Ans: d

3. **Which of the following keyword is used to overload an operator?**[L1]Page no:319

a. overload
b. operator
c.friend

d.overrider
Ans: b

4. **What will happen if a class is not having any name?**[L1]Page no:238
a. it cannot have a destructor
b. It cannot have a constructor.
c. It is not allowed.
d. Both A and B
Ans: d

5. Which inheritance type is used in the class given below? [L1] Page no:319

class A : public X, public Y
a. Multilevel inheritance
b. Multiple inheritance
c. Hybrid inheritance
d .Hierarchical Inheritance
Ans: b

6. Which of the following operators cannot be overloaded?[L1] Page no:319
a. []
b. ->
c. ?:
d. *
Ans:c

7. In which of the following a virtual call is resolved at the time of compilation?[L2] Page no:319
a. From inside the destructor.
b. From inside the constructor.
c. From inside the main *()*.
d. Both A and B.
Ans: d

8. Which of the following operator is overloaded for object cout?[L1] page no:319
a. >>
b. <<
c. +
d. =
Ans:b

9. Assume class TEST. Which of the following statements is/are responsible to invoke copy constructor? [L2] Page no:238
a. TEST T2 (T1)
b. TEST T4 = T1
c. T2 = T1
d. both a and b
Ans: d

10.Which of the following is the perfect set of operators that can't be overloaded in CPP? [L2]Page no:319
a. +=, ?, :: , >>
b. >>, <<, ?, *, sizeof()
c. :: , . , .* , ?:
d. :: , ->, * , new, delete
Ans: c

11. How many operators are supported by C++? [L1] page no:319

a. 30 operators
b. 40 operators
c. 45 operators
d. 65 operator

Ans:c

12. A non-member function that is given access to all members of a class within it is declared, is called[L1] Page no:319

a. Access function
b. Friend function
c. Operator functions
d. None of them

Ans:b

13. Which of the following operators should be preferred to overload as a global function rather than a member method?[L1] Page no:319

a. Postfix ++
b. Comparison Operator
c. Insertion Operator <<
d. Prefix++

Ans:c

14. We can overload which of the following C++ operators.[L2] Page no:319

a. Arithmetic operator (+, -, *, /)
b. Class Member Access Operators (., .*)
c. Size operator (sizeof)
d. Conditional operator (?:)

Ans: a

15. Operator overloading is also called …….. polymorphism.[L1] Page no:319

a. run time
b. initial time
c. Compile time
d. Completion time

Ans: c

16. Operator overloading is done with the help of a special function called ……, which describes the special task of an operator. [L1] Page no:319

a. overloading function
b.special task function
c. detail function
d. operator function

Ans:d

17. Overload an operator by naming it a[L1] Page no:319

a. variable

b. built-in type

c. function

d. class.

Ans:c

18. Which of the function operator cannot be over loaded[L2] Page no:319

a. <=

b. ?:

c. ==

d. *

Ans:b

19. Kind of diagrams which are used to show interactions between series of messages are classified as[L1] Page no:357

a. activity diagrams
b. state chart diagrams
c. collaboration diagrams
d. object lifeline diagrams

Ans:c

20. Dynamic aspects related to a system are shown with help of[L1] Page no:35

a. sequence diagrams
b. interaction diagrams
c. deployment diagrams

d. use case diagrams

Ans:b

4 Marks

21.Locate which of these are true with respect to the message arrows?[L2] Page no:357
a) The synchronous message arrow is used when a sending individual continues execution after sending the message
b) The asynchronous message arrow is used when a sending individual suspends execution after sending the message
c) **The dashed arrow is used either to show the return of control from a synchronous message or to create a new entity**
d) All of the mentioned
Ans:c

22.Recognize the following statements and find which is correct ?[L2]Page no:371
a)..Base class pointer cannot point to derived class.
b).**Derived class pointer cannot point to base class.**
c).Pointer to derived class cannot be created.
d).Pointer to base class cannot be created.
Ans:b

23.Implement and find the output of the following C++ code?[L3] Page no:319

```
#include <iostream>
#include <string>
using namespace std;
class Box{
        int capacity
public:
        Box(){}
        Box(double capacity){
                this->capacity = capacity;
        }
        bool operator<(Box b){
                return b.capacity < this->capacity? true : false;
        }
};

int main(int argc, char const *argv[])
{
        Box b1(10);
        Box b2 = Box(14);
        if(b1 < b2){
                cout<<"B1's capacity is small";
        }
        else{
                cout<<"B2's capacity is small";
        }
        return 0;}
```

a) B1's capacity is small
b) B2's capacity is small
c) Error
d) Segmentation fault
Ans:b

24.Solve the function prototype of the operator function which we need to define in this program so that the program has no errors?[L3] Page no:319

```
#include <iostream>
#include <string>
using namespace std;
class Box{
        int capacity;
public:
        Box(){}
        Box(double capacity){
                this->capacity = capacity;
        }
};
int main(int argc, char const *argv[])
{
        Box b1(10);
        Box b2 = Box(14);
        if(b1 == b2){
                cout<<"Equal";
        }
        else{
                cout<<"Not Equal";
        }
        return 0;
}
```

a)bool operator==();
b) bool operator==(Box b){}
c) bool operator==(Box b);
d) Box operator==();
Ans:c

25.Judge the operator to be overloaded in the following code to make the program error free?
[L5]Page no:319

```
#include <iostream>
#include <string>
using namespace std;
class Box{
        int capacity;
public:
        Box(){}
        Box(double capacity){
                this->capacity = capacity;
        }
```

```
};
int main(int argc, char const *argv[])
{
        Box b1(10);
        Box b2 = Box(14);
        if(b1 == b2){
                cout<<"Equal";
        }
        else{
                cout<<"Not Equal";
        }
        return 0;
}
```

a) +
b) ==
c) =
d) ()

Ans:b

26.Define the syntax of overloading operator + for class A?[L1] Page no:319
a) A operator+(argument_list){}
b) A operator[+](argument_list){}
c) int +(argument_list){}
d) int [+](argument_list){}

Ans:a

27.State which of the following statements is NOT valid about operator overloading?[L1] Page no:319
a)Only existing operators can be overloaded
b) The overloaded operator must have at least one operand of its class type
c) The overloaded operators follow the syntax rules of the original operator
d) The overloaded operator must not have at least one operand of its class type.

Ans:d

28. Identify the three different types of message arrows?[L2]Page no:357

a) Synchronous, asynchronous, asynchronous with instance creation
b) Self, Multiplied, instance generator
c) **Synchronous, Asynchronous, synchronous with instance creation**
d) None of the mentioned

Ans:c

29.Consider the declarations?[L1] Page no:319
char a;
const char aa = ‘h’;
char *na;

const char *naa;

Which of the following statements

Statement I: aa = a;

Statement II: na = &a;

Statement III: na = &aa;

is/are illegal?

a. Only I and II

b. Only II and III

c. **Only I and III**

d. All the three statements are illegal

Ans : c

30)Relate the differences between constructors and destructors and find which one is correct ? [L4] Page no:357

a. Constructors can take arguments but destructors cannot.

b. Constructors can be overloaded but destructors cannot be overloaded.

c. Destructors can take arguments but constructors cannot.

d. **Both (a) and (b)**

Ans:d

12 Marks

31.Investigate which property of Object Oriented Programming is exhibited by the code?[L6] page no:334

```
#include< iostream>
class quest
        {
          public:
        float add( float a, float b)
        {
          cout << "The sum is:";
           reurn(a+b); }  }
       int add( int a, int b, int c)
       {
 cout << "The sum is:";
          return(a+b+c);
        }

       int add( int a, int b=0)
        {
          cout << "The sum is:";
```

```
        return(a+b);
         }
      };
     int main( )

     {
quest q
        q. add( 50.5, 48.2)
        q. add( 42, 56, 82)
        q. add( 23)
         return 0;
        }
```

a. Compile time polymorphism
b. Run time polymorphism
c. polymorphism
d. virtual polymorphism

Ans:a

32.Implement and find the output of the following C++ code?
[L3]Pageno:319

```
#include <iostream>
 using namespace std;
 class sample
 {
    public:
        int x, y;
    sample() {};
        sample(int, int);
    sample operator + (sample);
      };
      sample::sample (int a, int b)
      {
        x = a;
        y = b;
      }
      sample sample::operator+ (sample param)
      {
        sample temp;
        temp.x = x + param.x;
        temp.y = y + param.y;
        return (temp);
      }
      int main ()
      {
        sample a (4,1);
        sample b (3,2);
```

```
    sample c;
    c = a + b;
    cout << c.x << "," << c.y;
    return 0;
}
```

a)5, 5
b) 7, 3
c) 3, 7
d) 3, 5
Ans:b

33.Judge the output of the following C++ code? [L5] Page no:319

```
#include <iostream>
using namespace std;
class Integer
{
    int i;
    public:
    Integer(int ii) : i(ii) {}
    const Integer
    operator+(const Integer& rv) const
    {
        cout << "operator+" << endl;
        return Integer(i + rv.i);
    }
    Integer&
    operator+=(const Integer& rv)
    {
        cout << "operator+=" << endl;
        i += rv.i;
        return *this;
    }
};
int main()
{
    int i = 1, j = 2, k = 3;
    k += i + j;
    Integer ii(1), jj(2), kk(3);
    kk += ii + jj;
}
```

a)
operator+
operator+=
b)
operator+=
operator+
c)
operator+
operator+

d)
operator+
operator=
Ans:a

34.Identify the output of the following C++code?[L2]Page no:319

```
#include <iostream>
using namespace std;
class myclass
{
   public:
   int i;
   myclass *operator->()
   {return this;}
};
int main()
{
   myclass ob;
   ob->i = 8;
   cout << ob.i << " " << ob->i;
   return 0;
}
```

a) 8 8
b) 11 11
c) 12 12
d) 0 0
Ans:A

35.Solve the following C++ code? [L3] Page no:319

```
#include <iostream>
#include <string>
using namespace std;
class A
{
      static int a;
   public:
      void show()
      {
            a++;
            cout<<"a: "<<a<<endl;
      }
};

int A::a = 5;

int main(int argc, char const *argv[])
{
      A a;
      return 0;
```

}

a)Error as a private member a is referenced outside the class

b) Segmentation fault

c) No output

d) Program compiles successfully but gives run-time error

Ans:c

36.Execute the following C++ code?[L3] Page no:319

```
#include <iostream>
#include <string>
using namespace std;
class A
{
        static int a;

  public:
        void show()
       {
                a++;
                cout<<"a: "<<a<<endl;
        }
        void operator.()
       {
                cout<<"Objects are added\n";
        }
};

class B
{
    public:
};

int main(int argc, char const *argv[])
{
        A a1, a2;
        return 0;
}
```

a)Run-time Error

b) Runs perfectly

c) Segmentation fault

d) Compile-time error

Ans:d

UNIT –III

1 Mark

1. **How many basic types of inheritance are provided as OOP feature?**[Page No: 371] [L1]
   a) 4
   b) 3
   c) 2
   d) 1
   Ans: a

2. **Which among the following best defines single level inheritance?**[Page No: 404] [L1]
   a) A class inheriting a derived class
   b) A class inheriting a base class
   c) A class inheriting a nested class
   d) A class which gets inherited by 2 classes
   Ans: b

3. **Which programming language doesn't support multiple inheritances?**[Page No: 404] [L1]
   a) C++ and Java
   b) C and C++
   c) Java and Small Talk
   d) Java
   Ans: d

4. **Which access type data gets derived as private member in derived class?**[Page No: 376] [L1]
   a) Private
   b) Public
   c) Protected
   d) Protected and Private
   Ans: a

5. **How can you make the private members inheritable?**[Page No: 396] [L1]
   a) By making their visibility mode as public only
   b) By making their visibility mode as protected only
   c) By making their visibility mode as private in derived class
   d) It can be done both by making the visibility mode public or protected
   Ans: d

6. **Which problem arises due to multiple inheritances, if hierarchical inheritance is used previously for its base classes?**[Page No: 406] [L1]
   a) Diamond
   b) Circle
   c) Triangle
   d) Loop
   Ans: a

7. **How many classes should a program contain to implement the multiple inheritance?**

[Page No: 410] [L1]
a) Only 1
b) At least 1
c) At least 3
d) Exactly 3
Ans: c

8. **Is it compulsory to have constructor for all the classes involved in multiple inheritance?**
[Page No: 414] [L1]
a) Yes, always
b) Yes, only if no abstract class is involved
c) No, only classes being used should have a constructor
d) No, they must not contain constructors
Ans: b

9. **Can the derived class be made abstract if multiple inheritances is used?**[Page No: 407] [L1]

a) No, because other classes must be abstract too
b) Yes, if all the functions are implemented
c) Yes, if all the methods are predefined
d) No, since constructors won't be there
Ans: d

10. **Name the function whose definition can be substituted at a place where its function call is made: ?**
[Page No: 164] [L1]

a) friends function
b) inline function
c) volatile function
d) external function

Ans: b

11. **Which keyword is used to declare the friend function?**[Page No: 520] [L1]

a) friend
b) friend
c) classfriend
d) myfriend
Ans: b

12. **Which of the given modifiers can be used to prevent Method overriding?** [Page No: 371] [L1]

a) Static
b) Constant
c) Sealed
d) final
Ans: c

13. Which of the following cannot be used to declare a class as a virtual?[Page No: 504] [L1]

a) Methods
b) Properties
c) Events
d) Fields
Ans: d

14. Can abstract class have main () function defined inside it?[Page No: 510] [L1]

a) Yes, depending on return type of main()
b) Yes, always
c) No, main must not be defined inside abstract class
d) No, because main() is not abstract function
Ans: b

15. If there is an abstract method in a class then, _____[Page No: 511] [L1]

a) Class must be abstract class
b) Class may or may not be abstract class
c) Class is generic
d) Class must be public
Ans: a

16. Which of the following UML diagrams has a static view?[Page No: 160] [L1]

a) Collaboration
b) Use case
c) State chart
d) Activity
Ans: b

17. Which diagram in UML shows a complete or partial view of the structure of a modeled system at a specific time?[Page No: 162] [L1]

a) Sequence Diagram
b) Collaboration Diagram
c) Class Diagram
d) Object Diagram
Ans: d

18. Use of pointers or reference to an abstract class gives rise to which among the following feature? [Page No: 512] [L1]

a) Static Polymorphism
b) Runtime polymorphism
c) Compile time Polymorphism
d) Polymorphism within methods
Ans: b

19. Activity diagram, use case diagram, collaboration diagram and sequence diagram are considered as types of ?[Page No: 186] [L1]

a) non-behavioral diagrams
b) nonstructural diagrams
c) structural diagrams
d) behavioral diagrams

Ans: d

20. Diagrams which are used to distribute files, libraries and tables across topology of hardware are called?[Page No: 188] [L1]

a) deployment diagrams
b) use case diagrams
c) sequence diagrams
d) collaboration diagrams

Ans: d

4 Marks

21. Which among the following is correct for multiple inheritance? [Page No: 405] [L2]
a) class student{public: int marks;}s; class stream{int total;}; class topper:public student, public stream{ };
b) class student{int marks;}; class stream{ }; class topper: public student{ };
c) class student{int marks;}; class stream:public student{ };
d) class student{ }; class stream{ }; class topper{ };

Ans: a

22.While inheriting a class, if no access mode is specified, then which among the following is true? (in C++) [Page No: 398] [L1]
a) It gets inherited publicly by default
b) It gets inherited protected by default
c) It gets inherited privately by default
d) It is not possible

Ans: c

23. What is the output of this program? [Page No: 520] [L3]

```
#include <iostream>
using namespace std;
class sample
{
   private:
   int a, b;
   public:
   void test()
   {
      a = 100;
      b = 200;
```

```
    }
    friend int compute(sample e1);
};
int compute(sample e1)
{
    return int(e1.a + e1.b) - 5;
}
int main()
{
    sample e;
    e.test();
    cout  << compute(e);
    return 0;
}
```

a) 100
b) 200
c) 300
d) 295
Ans: d

24. Point out the error (if any) in the code shown below: [Page No: 196] [L3]

```
#include <stdio.h>
void inline func1(float b)
{
   printf ("%lf\n",b*2);
}
int main()
{
    inline func1(2.2);
    return 0;
}
```

a) No error
b) Error in statement: void inline func1(float b)
c) Error in statement: printf(“%lf\n”,b*2);
d) Error in statement: inline func1(2.2);

Ans: d

25. What is the output of the code shown below? [Page No: 522] [L2]

```
#include <stdio.h>
void inline func1(char b[10])
{
   printf ("%c\n",b[2]);
}
int main()
{
    func1("sanfoundry");
    return 0;
}
```

a) s

b) n
c) a
d) error

Ans: b

26. What is the output of this program? [Page No: 518] [L3]

```
#include <iostream>
using namespace std;
class sample
{
   private:
   int a, b;
   public:
   void test()
   {
      a = 100;
      b = 200;
   }
   friend int compute(sample e1);
};
int compute(sample e1)
{
   return int(e1.a + e1.b) - 5;
}
int main()
{
   sample e;
   e.test();
   cout  << compute(e);
   return 0;
}
```

a) 100
b) 200
c) 300
d) 295
Ans: d

27. What is the output of this program? [[Page No: 395] L3]

```
#include <iostream>
using namespace std;

class A
{
public:
   virtual void fun() { cout << "A::fun() "; }
};

class B: public A
{
```

```
public:
   void fun() { cout << "B::fun() "; }
};

class C: public B
{
public:
   void fun() { cout << "C::fun() "; }
};

int main()
{
   B *bp = new C;
   bp->fun();
   return 0;
}
```

a) A::fun() b) B::fun() c) C::fun() d)A:B:C:fun()

Ans: c

28. What type of core-relationship is represented by the symbol in the figure below? [Page No:195] [L1]

a) Aggregation
b) Dependency
c) Generalization
d) Association

Ans: a

29. Which core element of UML is being shown in the figure? [Page No: 280] [L1]

a) Node
b) Interface
c) Class
d) Component

Ans: d

30. If class A inherits class B and class C as "class A: public class B, public class C {// class body ;};", which class constructor will be called first? [Page No: 397] [L2]

a) Class A
b) Class B
c) Class C
d) All together

Ans: b

12 Marks

31. (i) When the inheritance is private, the private methods in base class are _____ in the derived class. [Page No: 396] [L2]

a. inaccessible
b.Accessible
c. Protected
d. Public

(ii) What will be the order of execution of base class constructors in the following method of inheritance?

class a: public b, public c {….};

a. b(); c(); a();
b. c(); b(); a();
c. a(); b(); c();
d. b(); a(); c();

iii) What are the things are inherited from the base class?

a. Constructor and its destructor
b. Operator= () members
c. Friends
d. All of the above

Ans: (i)a (ii)a (iii)d

32.Consider the following code [Page No: 399] [L3]

```
class a1 {
  public:
    ~a1()  { cout << " a1" << endl; }
};

class a2 {
  public:
    ~a2()  { cout << " a2" << endl; }
};

class c: public a1, public a2 {
  public:
    ~derived()  { cout << " derived" << endl; }
};

int main()
{
  Derived d;
  return 0;

}
```

(i) The output of the above code is

a. a1
   a2
   derived
b. derived
   a2
   a1
c .derived
d. a2

(ii) Which design patterns benefit from the multiple inheritances?

a. Adapter and observer pattern
b.Code pattern
c. Glue pattern
d. None of the mentioned

(iii) What is the ability to group some lines of code that can be included in the program?

a) Specific task
b) program control
c) modularization
d) macros

Ans: (i) b (ii) a (iii) c

33.Consider the following code [Page No: 382] [L3]

```
include<iostream>
using namespace std;

class Base
{
public:
   virtual void show() { cout<<" In Base \n"; }
};

class Derived: public Base
{
public:
   void show() { cout<<"In Derived \n"; }
};

int main(void)
{
   Base *bp = new Derived;
   bp->show();
```

```
    Base &br = *bp;
    br.show();

    return 0;
}
```

(i) The o/p of above code is

a) in base in base b) in base in derived c) in derived in derived d) in derived in base

(ii) Can static functions be virtual? Will the following program compile?

```
#include<iostream>
using namespace std;

class Test
{
  public:
    virtual static void fun()  { }
};
```

a) Yes, virtual b) Not a virtual c) static d) based on compiler

(iii) Which is used to create a pure virtual function?

a) $
b) =0
c) &
d)!

(iv) Which is also called as abstract class?

a) Virtual function
b) pure virtual function
c) derived class
d) base class

Ans: (i)c (ii) a (iii) b (iv) b

34.**(i) The element of state chart diagrams which uses round corner boxes to represent situations of an object is classified as** [Page No: 220] [L2]

a) Lifeline marker b) iteration marker c) transitions d) state

(ii) The relationship between two independent data table is classified as

a)Structural relationship b) Non structural relationship c) identifying relationship d) Non identifying relationship

(iii) In state chart diagrams, the element which is shown with the help of double line filled circle with pointing arrow is classified as

a) Initial state b) Two degree state c) Final state d) Zero degree state

(iv) In state chart diagrams, the element which is shown with the help of solid circle with outgoing arrow is classified as

a) Initial state b) Two degree state c) Final state d) Zero degree state

Ans: (i) d (ii) d (iii)c (iv) a

35. (i) The elements included in state chart diagrams are [Page No: 220] [L2]
a) Transitions b) Condition marker c) lifeline marker d) iteration marker

(ii) Which of the following determines the state diagram?
a) The UML notation for specifying finite automata is the state diagram
b) In state diagrams, states are represented by rounded rectangles
c) All of the mentioned
d) None of the mentioned

(iii) Which of the statements state the name compartment?
a) The first compartment is the name compartment
b) It contains the state name; State names are optional and may be path names
c) The name compartment can never be omitted
d) The first compartment is the name compartment, It contains the state name; State names are optional and may be path names

Ans: (i)a (ii)c (iii)d

36. (i) If class Apple inherits class Banana and class Cat as "class Apple: public class Banana, public class Cat {// class body ;}; ", which class constructor will be called first?[Page No: 403] [L3]
a) Class Apple
b) Class Banana
c) Class Cat
d) All together

(ii) Why does diamond problem arise due to multiple inheritance?
a) Methods with same name creates ambiguity and conflict
b) Methods inherited from the super class may conflict
c) Derived class gets overloaded with more than two class methods
d) Derived class can't distinguish the owner class of any derived method

(iii) When multiple inheritances is used, which class object should be used in order to access all the available members of parent and derived class?
a) Derived class object
b) Parent class objects
c) Use Abstract derived class
d) Derive a class from derived class

(iv)Which members can't be accessed in derived class in multiple inheritance?
a) Private members of base
b) Public members of base

c) Protected members of base
d) All the members of base

Ans: (i) b(ii)a (iii)a (iv)a

UNIT-IV

1 MARK

1. The STL can be used as a standard approach for----------[L1, R5-726]
a) Storing and sorting
b) Storing and processing data
c) data processing only
d) storing only
Ans:b
2. Name the Container which uses both stack and queue.[L1, R5-728]
a) storage
b) linked list
c) queuing
d) Deque
Ans:d
3. Identify the characteristics of vector container.[L2, R5-728]
a) Relocating, expandable array
b) Fixed size
c) Doubly linked list
d) link vector
Ans:a
4. Associative container uses---------to access data.[L1, R5-729]
a) queue
b) Keys
c) stack
d) string
Ans:b
5. Class templates are generally used for--------[L1, R5-690]
a) Data storage
b) debug
c) fixed data type
d) storage
Ans:a
6. In UML, Templates are also called as--------[L1, R5-702]
a) container

b) modified
c) Parameterized
d) generic
Ans:c
7. ___ __ specifies additional detail about UML element[L1, R5-703]
a) Stereotype
b) container
c) associative container
d) data processing
Ans:a

8.-------- is visible only to its containing package and to its nested package.[L1, R1-163]
a) protected
b) public
c) Private
d) package
Ans:c
9. Notation is used to specify the required and provided interfaces of the components. The interfaces between the components are named as------[L1, R1-172]
a) Assembly connectors
b) cooling controllers
c) Environmental controller
d) Plan analyst
Ans:a

10. List the 3 essential elements of a deployment diagram?[L1, R1-177]
a) Artifacts, nodes and connections.
b) stack, queue, deque
c) memory, database, connections
d) package, element, deployment
Ans:a

11. Activity, use case diagram, collaboration diagram and sequence diagram are categorized as [L4, R1-147]
a) non-behavioral diagrams
b) non structural diagrams
c) structural diagrams
d) Behavioral diagrams
Ans:d

12. Recognize which diagram is used to distribute files, libraries and tables across topology of hardware?
[L1, R1-171]
a) Deployment diagrams
b) use case diagrams
c) sequence diagrams
d) collaboration diagrams
Ans:a

13. List the essentials in package diagram [L1, R1-165]
a) Package notation, element visibility, dependency relationship
b) package notation, sequence, dependency relationship
c) Dependency, element visibility
d) package, deployment, sequence
Ans:a

14. Good packages are---------coupled and highly cohesive among the elements in package. [L1, R1-167]

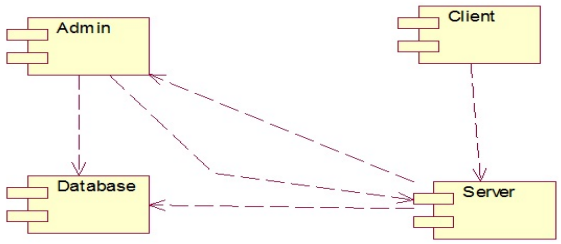
a) Tightly
b) highly
c) loosely
d) semi
Ans:c

15. Identify the validity of template parameters? [L1, R5-682]
a) inside that block only
b) inside the class
c) whole program
d) inside the main class
Ans:a

16. Identify the core element of UML in the below figure?[L2, R1-163]

a) Node
b) Interface
c) Class
d) Component
Ans:d

17. Recognize the UML diagram shown below? [L2, R1-163]



a) Component
b) Deployment
c) Use Case
d) DFD
Ans:a

18. ----------type of program can be included in try block? [L1, R5-705]
a) static memory allocation
b) Dynamic memory allocation
c) const reference
d) pointer
Ans:b

19. ---------statement is used to catch all types of exceptions. [L1, R5-705]
a) catch()
b) catch(Test t)
c) catch(…)
d) none of the mentioned
Ans:c

20. The ---------- class name must be included in the class in which it is located. [L1, R5-709]
a) try
b) Exception
c) catch
d) template
Ans:b

4 Mark

21. From where does the template class derived? [L2, R5-702]

1. Regular non-templated C++ class
2. Templated class
3. A or B
4. None of the above

a) Only 1st is correct
b) Only 2nd is correct
c) Both 1st and 2nd are correct
d) Maybe 1st or 2nd is correct.
Ans:d

22. Explore the correct statement about string template?[L3, R5-683]
a) It is used to replace a string.
b) It is used to replace a string with another string at runtime.
c) It is used to delete a string.
d) None of the above
Ans:b

23. Identify which among the following is not correct. [L2, R5-683]
a) **template <class T> func(T x) {}**
template <class T> func<T*>(T* x) {}
b) template <class T>
class myObject {};
c) template <class T>
class myObj { template <class R> memFunc() {} };
d) All of the above are correct.
Ans:a

24. Examine whether templates are conceptually related to polymorphism? [L4, R5-682]
a) Not Related
b) Only when the template types are objects
c) Yes, but compile-time polymorphism
d) Yes, but run-time polymorphism
Ans:c

25. Identify an invalid template declaration.[L2, R5-683]
a) template <int x> int func() {return x;}
b) template <double x> double func() {return x;}
c) template <typename x> void func(x t) {}
Ans:b

26. Relate an option to restrict a function to throw certain exceptions?[L3, R5-705]
a) Defining multiple try and catch block inside a function
b) Defining generic function within try block
c) Defining function with throw clause

d) It is not possible in CPP to restrict a function
Ans:c

27. Select the ways to represent nodes in a deployment diagram? [L6, R1-171]
a) Nodes instances are underlined identifiers of the form name:type
b) The name may be left off, indicating an unnamed instance of the type
c) The type may be left off, indicating a named instance with an unspecified type
d) All of the mentioned
Ans:d

28. Examine: In component diagrams, building block which is represented with two rectangles laid on left side is classified as [L4, R1-163]
a) type of components
b) interfaces
c) dependency relationships
d) State dependency
Ans:c

29. Choose the ways to represent nodes in a deployment diagram? [L6, R1-171]
a) Nodes instances are underlined identifiers of the form name:type
b) The name may be left off,indicating an unnamed instance of the type
c) The type may be left off, indicating a named instance with an unspecified type
d) All of the mentioned
Ans:d

30. Identify the ways to represent nodes in a deployment diagram? [L2, R1-171]
a. Nodes instances are underlined identifiers of the form name:type
b. The name may be left off,indicating an unnamed instance of the type
c. The type may be left off, indicating a named instance with an unspecified type
d. All of the mentioned
Ans:d

12 Mark

31. Predict the output of the following C++ code? [L5, R5-682]

```
#include <iostream>
using namespace std;
template <class T, int N>
class mysequence
{
   T memblock [N];
   public:
   void setmember (int x, T value);
   T getmember (int x);
};
template <class T, int N>
void mysequence<T,N> :: setmember (int x, T value)
{
   memblock[x] = value;
}
template <class T, int N>
```

```
T mysequence<T,N> :: getmember (int x)
{
   return memblock[x];
}
int main ()
{
   mysequence <int, 5> myints;
   mysequence <double, 5> myfloats;
   myints.setmember (0, 100);
   myfloats.setmember (3, 3.1416);
   cout << myints.getmember(0) << '\n';
   cout << myfloats.getmember(3) << '\n';
   return 0;
}
```

a) 100
b) 3.1416
c) 100
3.1416
d) 4.14
Ans:c

32. Identify the best-defined syntax for template function? [L4, R5-684]

1. template return_type Function_Name(Parameters)
2. template
   return_type Function_Name(Parameters)
3. tmp return_type Function_Name(Parameter1, parameter 2)
4. template
   Return_type(parameters)

a) Only 1st syntax is correct
b) Only 2nd syntax is correct
c) Both 1st and 2nd syntaxes are correct
d) None of the above
Ans:c

33. Predict the output of the following C++ code? [L5, R5-682]

```
#include <iostream>
using namespace std;
template <class T>
T max (T& a, T& b)
{
   return (a>b?a:b);
}
int main ()
{
   int i = 5, j = 6, k;
   long l = 10, m = 5, n;
   k = max(i, j);
   n = max(l, m);
   cout << k << endl;
   cout << n << endl;
   return 0;
}
```

a) 6

b) 6
10
c) 5
10
d) 5
Ans:b

34. Predict the output of this program? [L5, R5-682]

```
#include < iostream >
#include < string >
using namespace std;
template < typename T >
void print_mydata(T output)
{
cout << output << endl;
}
int main()
{
double d = 5.5;
string s("Hello World");
print_mydata( d );
print_mydata( s );
return 0;
}
```

a) 5.5
b) 5.5 Hello World
c) Hello World
d) None of the above
Ans:b

35. Predict the output of this program? [L5, R5-705]

```
#include <iostream>
   using namespace std;
   int main()
   {
      int x = -1;
      try
      {
         if (x < 0)
         {
            throw x;
         }
         else
         {
            cout<<x;
         }
      }
      catch (int x )
      {
         cout << "Exception occurred: Thrown value is " << x << endl;
      }
      return 0;
   }
```

a) -1

b) 0
c) Exception occurred: Thrown value is -1
d) Error
Ans:c

36. Predict the output of this program? [L5, R5-705]

```
#include <iostream>
   #include <exception>
   using namespace std;
   void myunexpected ()
   {
      cout << "unexpected called\n";
      throw 0;
   }
   void myfunction () throw (int)
   {
      throw 'x';
   }
   int main ()
   {
      set_unexpected (myunexpected);
      try
      {
         myfunction();
      }
      catch (int)
      {
         cout << "caught int\n";
      }
      catch (...)
      {
         cout << "caught other exception\n";
      }
      return 0;
   }
```

a) caught other exception
b) caught int
c) unexpected called
d) both caught int & unexpected called
Ans:d

UNIT V

1 Mark

1. What kind of library is Standard Template Library? [Page No: 620] [L1]

a) Polymorphic

b) Generic

c) Both Polymorphic & Generic

d) None of the mentioned

Ans:b

2. To what type of object does the container can be instantiated? [Page No: 650] [L1]

a) int

b) float

c) double

d) any type of object

Ans:d

3. What type of class template is list? [Page No: 679] [L1]

a) Class-based

b) Node-based

c) Method-based

d) None of the mentioned

Ans:b

4. What type of access does deque and vector provide? [Page No: 675] [L1]

a) Linear access

b) Parallel access

c) Random access

d) None of the mentioned

Ans:c

5. Where does the vector add the item? [Page No: 685] [L1]

a) End

b) Insert

c) Middle

d) None of the mentioned

Ans:a

6. Which are not full container classes in C++?[Page No: 655] [L1]

a) Sequence container

b) Associative container

c) Container adaptor

d) None of the mentioned

Ans:c

7. What is the lifetime of the element in container? [Page No: 660] [L1]

a) Whole program

b) Outside the block

c) Everywhere

d) Only on that container

Ans:d

8. Which operator is used to insert the data into file? [Page No: 705] [L1]

a) >>

b) <<

c) <

d) None of the mentioned

Ans:b

9. Which function is used to position back from the end of file object? [Page No: 725] [L1]

a) seekg

b) seekp

c) both seekg&seekp

d) none of the mentioned

Ans:a

10. How many objects are used for input and output to a string? [Page No: 712] [L1]

a) 1

b) 2

c) 3

d) 4

Ans:c

11. Which is used to handle the exceptions in c++?[Page No: 745] [L1]

a) catch handler

b) handler

c) exception handler

d) none of the mentioned

Ans: c

12. Which type of program is recommended to include in try block? [Page No: 715] [L1]

a) static memory allocation

b) dynamic memory allocation

c) const reference

d) pointer

Ans:b

13. Which statement is used to catch all types of exceptions? [Page No: 706] [L1]

a) catch()

b) catch(Test t)

c) catch(…)

d) none of the mentioned

Ans:c

14. How to handle error in the destructor? [Page No: 710] [L1]

a) throwing

b) terminate

c) both throwing & terminate

d) none of the mentioned

Ans:b

15. What kind of exceptions are available in c++?[Page No: 725] [L1]

a) handled

b) unhandled

c) static

d) dynamic

Ans:b

16. What do associate containers implement? [Page No: 670] [L1]

a) Arrays

b) Associative arrays

c) Functional Arrays

d) Static arrays

Ans: b

17. By using which of the following the elements in the associate container can be efficiently accessed? [Page No: 656] [L1]

a) Key

b) Position

c) Both Key & Position

d) Value

Ans: a

18. How many items are presented in the associate container? [Page No: 658] [L1]

a) 2

b) 3

c) 4

d) 5

Ans: c

19. What are the containers? [Page No: 658] [L1]

a) Containers store objects and data

b) Containers stores all the algorithms

c) Containers contain overloaded functions

d) Containers contain set of Iterators

Ans: a

20. In how many categories, containers are divided? [Page No: 660] [L1]

a) 1

b) 2

c) 3

d) 4

Ans:d

4 Marks

21.What are the Sequence Containers? [Page No: 675] [L2]

a) Containers that implements data structures which can be accessed sequentially

b) Containers that implements sorted data structures for fast search in O(logn)

c) Containers that implements unsorted(hashed) data structures for quick search in O(1)

d) Containers that implements data structures which can be accessed non-sequentially

Ans:a

22.How many Sequence Containers are provided by C++?[Page No: 671] [L2]

a) 2

b) 3

c) 4

d) 5

Ans:d

23.What are the Associative Containers? [Page No: 667] [L2]

a) Containers that implements data structures which can be accessed sequentially

b) Containers that implements sorted data structures for fast search in O(logn)

c) Containers that implements unsorted(hashed) data structures for quick search in O(1)

d) Containers that implements data structures which can be accessed non-sequentially

Ans:b

24. What are Container Adaptors? [Page No: 671] [L2]

a) Containers that implements data structures which can be accessed sequentially

b) Containers that implements sorted data structures for fast search in O(logn)

c) Containers that implements unsorted(hashed) data structures for quick search in O(1)

d) Containers that provide a different interface for sequential containers

Ans:d

25.What are Iterators? [Page No: 740] [L2]

a) Iterators are used to iterate over C-like arrays

b) Iterators are used to iterate over pointers

c) Iterators are used to point memory addresses of STL containers

d) Iterators are used to iterate over functions

Ans:c

26. How many types of Iterators are provided by C++?[Page No: 745][L2]

a) 2

b) 3

c) 4

d) 5

Ans:d

27. Which of the following statements are correct? [Page No: 654] [L2]

a) It is not possible to combine two or more file opening mode in open() method.

b) It is possible to combine two or more file opening mode in open() method.

c) ios::in and ios::out are input and output file opening mode respectively.

Ans:a

28. Which function is used to reposition the file pointer? [Page No: 709] [L2]

a) moveg()

b) seekg()

c) changep()

d) go_p()

Ans:b

29. Where is a file temporarily stored before read or write operation in C language.? [Page No: 723] [L2]

A) Notepad
B) RAM
C) Hard disk
D) Buffer

Ans:d

30. What is the use of ios::trunc mode? [Page No: 725] [L2]

a) To open a file in input mode

b) To open a file in output mode

c) To truncate an existing file to half

d) To truncate an existing file to zero Ans:d

12 Marks

31.Merge sort[Page No: 780] [L2]

i)Merge sort uses which of the following technique to implement sorting?

a) backtracking

b) greedy algorithm

c) divide and conquer

d) dynamic programming

ii)What is the worst case time complexity of merge sort?

a) O(n log n)

b) $O(n^2)$

c) $O(n^2 \log n)$

d) $O(n \log n^2)$

iii) Which of the following method is used for sorting in merge sort?

a) merging

b) partitioning

c) selection

d) exchanging

iv) Which of the following is not a variant of merge sort?

a) in-place merge sort

b) bottom up merge sort

c) top down merge sort

d) linear merge sort

Answer

i)c

ii)a

iii)a

iv)d

32.File handling[Page No: 726] [L2]

i) Which header file is required to use file I/O operations?

a) <ifstream>

b) <ostream>

c) <fstream>
d) <iostream>

ii) Which of the following is used to create an output stream?
a) ofstream
b) ifstream
c) iostream
d) fsstream

iii) Which of the following is used to create a stream that performs both input and output operations?
a) ofstream
b) ifstream
c) iostream
d) fstream

iv) Which of the following is not used as a file opening mode?
a) ios::trunc
b) ios::binary
c) ios::in
d) ios::ate

Answer

i)c

ii)a

iii)d

iv)a

33. What will be the output of the following C++ code? [Page No: 715] [L3]

i)
```
#include <iostream>
using namespace std;
int main ()
{
    char first, second;
    cout << "Enter a word: ";
    first = cin.get();
    cin.sync();
```

```
    second = cin.get();
    cout << first << endl;
    cout << second << endl;
    return 0;
}
```

a)first
b)second
c)returns first 2 letter or number from the entered word
d) third

ii) How many objects are used for input and output to a string?
a) 1
b) 2
c) 3
d) 4

iii) Which member function is used to determine whether the stream object is currently associated with a file?
a) is_open
b) buf
c) string
d) is_out

iv) Which function is used to position back from the end of file object?
a) seekg
b) seekp
c) both seekg & seekp
d) seekf

Answer

i)c

ii)c

iii)a

iv)a

34. Iterators[Page No: 749] [L2]

i) How many categories of iterators are there in c++?
a) 2
b) 4

c) 5
d) 3

ii) What are Iterators?
a) STL component used to point a memory address of a container
b) STL component used for vectors
c) STL component used to call functions efficiently
d) STL component used to define template classes

iii) Which function is used increment the iterator by a particular value?
a) next()
b) advance()
c) prev()
d) move()

iv) Pick the correct statement.
a) Input iterator moves sequentially forward
b) Input iterator moves sequentially backward
c) Input iterator moves in both direction
d) Input iterator moves sequentially downwards

Answer

i)c

ii)a

iii)b

iv)a

35. Stack[Page No: 734] [L2]

i) Consider the usual algorithm for determining whether a sequence of parentheses is balanced.
The maximum number of parentheses that appear on the stack AT ANY ONE TIME when the algorithm analyzes: (()(())(())) are:
a) 1
b) 2
c) 3
d) 4 or more

ii) Process of removing an element from stack is called _____
a) Create
b) Push
c) Evaluation
d) Pop

iii) Pushing an element into stack already having five elements and stack size of 5, then stack becomes
a) Overflow
b) Crash
c) Underflow
d) User flow

iv) Which of the following applications may use a stack?
a) A parentheses balancing program
b) Tracking of local variables at run time
c) Compiler Syntax Analyzer
d) Data Transfer between two asynchronous process

Answer

i)c

ii)d

iii)a

iv)d

36. Associative Container[Page No: 662] [L2]
i) What do associate containers implement?
a) Arrays
b) Associative arrays
c) Functional Arrays
d) Static arrays
ii) By using which of the following the elements in the associate container can be efficiently accessed?
a) Key
b) Position
c) Both Key & Position
d) Value

iii) How many items are presented in the associate container?
a) 2
b) 3
c) 4
d) 5

iv) How many instances are allowed by map and set while inserting an element into container?
a) 1
b) 2
c) 3
d) Multiple

Answer i)b ii)a iii)c iv)a