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

PART A (1 Mark)

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

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

PART- B (4 Marks)

1. How can we prevent a class from instantiation?
2. Construct Use-case diagram for an Online Shopping Application
3. List out the difference between procedure oriented programming & Object oriented programming. (CLA1-L1)
4. Write syntax of class, objects and methods and explain with example. (CLA1-L1)
5. Differentiate public and private functions with an example. (CLA1-L2)
6. Give example for static keyword and explain. (CLA1-L2)
7. Write a C++ program to generate factorial of a number using class (CLA1-L3)
8. Write an example program to demonstrate type conversions and explain.(CLA1-L3)
9. List the features of OOPS and explain. (CLA1-L1)
10. Write the drawbacks in procedural oriented programming. (CLA1-L1)
11. Differentiate class and objects with example. (CLA1-L2)
12. Give an example for pointer and explain. (CLA1-L2)
13. Write a C++ program to generate Fibonacci series using class (CLA1-L3)
14. Write an example program to demonstrate casting (CLA1-L3)
15. Illustrate the relationship used in Use case.
16. What is UML? List out the UML Diagrams

17. Compare Include and Extend use case relationships.
18. When to use class diagram?
19. Explain the terms: Polymorphism and Encapsulation
20. Consider a Banking System. Identify three entities in the system which can be represented using classes and show the relationship between them using UML class diagrams
21. Explain how objects are passed as function parameters with a suitable example.
22. What is a constructor? What are its uses?
23. Draw the Use Case diagram of online railway ticket reservation system

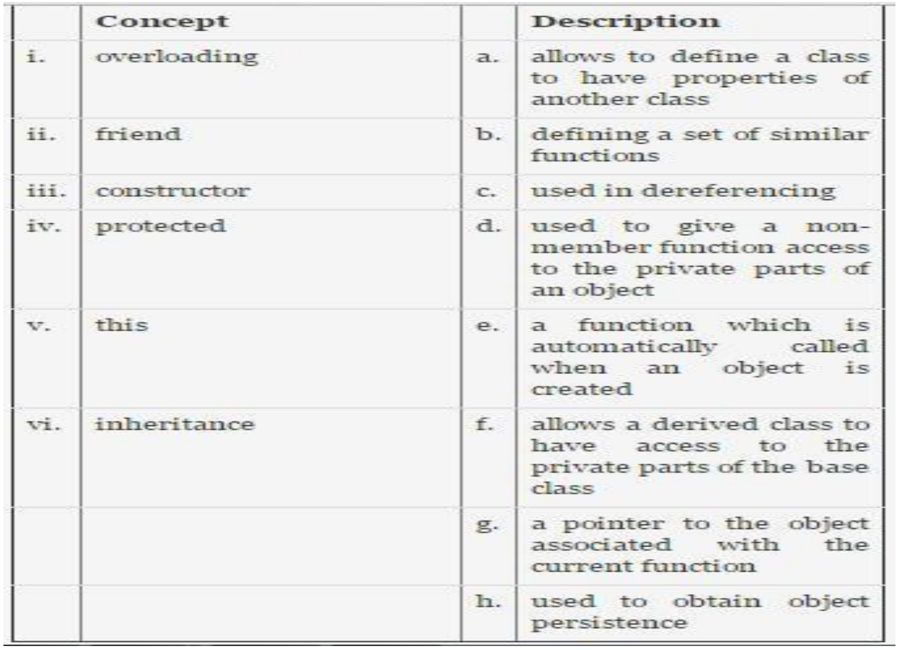
24. What will be the output of the following code? (L2) (Page no 36)

```
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();
}
```

ANS inside constructor something destroying 2this is i in main destroying 1 (Explain)

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. **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;
}
```

28. **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;
}
```

Ans: Compiler Error (Explain)

29.

PART C (12 Marks)

Answer the Questions Briefly:-

1. Write the problem statement for Library Management system. Design UML Class diagram and explain its components (CLA1-L3)

2. What is UML? Explain its components with suitable example (CLA1-L1)

3. There are 50 computers available in computer programming lab where each computers are used six hours per day. Write a C++ program using classes and objects that contain getDetail() for getting input from user,calculatesecondperDay() for calculating the usage of each computer in seconds per day

,calculateminutesperWeek() for calculating the usage of each computer in minutes per week ,calculatehourperMonth() for calculating usage of each computer in hour per month and calculatedayperYear() for calculating usage of each computer in day per year (CLA1-L3)

4. Give example for cast? Explain OOPS features with suitable example.(CLA1-L2)

5. Write the problem statement for Railway Reservation System. Design UML Class diagram and explain its components (CLA1-L3)

6. Explain in detail about constructor and demonstrate with an example (CLA1-L1)

7. Design UML Class diagram for ATM system and write C++ code for each class. (CLA1-L3)

8. Compare local and global variable and demonstrate classes and objects with real time example (CLA2-L2)

9. Create three classes with names Shape, Rectangle and Circle and make use of the functions getdata(), printdata(), and area(). To find the area of circle and rectangle, which type of inheritance is suitable? Why? Explain?

10. A Library lends books and magazines to member, who is registered in the system. It also maintains the purchase of new books and magazines for the Library. A member can reserve a book or magazine that is not currently available in the library, so that when it is returned or purchased by the library, that person is notified. The library can easily create, replace and delete information about the books, members, and reservation in the system. The books transactions are stored in the database. The fine list while the member returns the book after the due date must be
generated. Design the use case diagram and discover the users and actors of this system, and the interactions between them must be depicted.

11. Design and illustrate the use case model for activities involved in ordering food in a restaurant from the point when the customer enters a restaurant to the point when he leaves the restaurant.

12. Explain the benefits and concepts of use case and use case model and analyze the relating use cases have in ATM system

13. Describe a suitable example showing the various relationships used in Use Case and also give a short note on each relationship.

14. List the Various UML diagrams and explain the purpose of each diagram

15. By considering the Library management system, perform the object oriented System Development and give the use case model for the same (use include, extend and generalization)

16. A University conducts examinations and the results are announced. Prepare a report for the following:

- Print the marks in the register number order semester wise for each department
- Print the Arrear list semester wise.
- Prepare a Rank list for each department.

- Prepare the final aggregate mark list for final year students.

Identify the problem statement and Design and Explain the classes for each sequence. Design the Use case case & Class diagram

17. Analyze and design for Library Information System which comprises the following notations and explain them. (i)Aggregation (ii) Composition (iii) Association.

18. Design the Class diagram for Hospital management system?

19. With a suitable example explain how to design a class. Give all possible representation in a class (such as: name, attribute, visibility, methods, and responsibilities).

20. Draw Use case Diagram for online Pizza ordering system

UNIT 2:
PART -A

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
A. A class can have more than one constructor.
B. They can be inherited.
C. Their address can be referred.
D. Constructors cannot be declared in protected section of the class.
E. Constructors cannot return values.
a. Only A, B, D
b. A,B,D,E
c. A,C,E
d. A,D,E
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:

1. What is the necessity of constructor overloading?
2. Categorize the types of Constructors
3. Give the definition of Copy Constructors with example
4. Explain the various modes of inheritance with example
5. Define function overloading with example
6. Define method overloading and Write a program to implement method overloading with different number of arguments and same return types
7. Write a program to implement method overloading with different number of arguments and same as well as different return types
8. Write down the restrictions on Operator overloading
9. Define operator overloading with syntax and example
10. List out the limitations of operator overawing.
11. Define overloading unary and binary operators with an example
12. Explain overloading assignment operators with an example
13. Define inheritance and its types
14. What is the benefit of using inheritance?
15. Define interaction diagram
16. Define sequence diagram

17. Define collaboration diagram with its notation
18. Explain the modes of inheritance with an example
19. Define static constructor
20. Relate the differences between constructors and destructors
21. List out the types of Interaction diagram and Notations
22. 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;
}
```

Ans
operator+
operator+= (Justify your answer)

21.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;
}
```

Ans: 8 8 (Justify your answer)

22. Can we have virtual destructors? If so what is the use of virtual destructors.

Ans. Yes, we can. This is to make sure that the correct class destructor is called at run time. Specifically when we use base class pointer or reference to hold the derived class object. If we don’t have virtual destructor, then it will end up in calling only base class destructor.

```
// CPP program without virtual destructor
// causing undefined behavior
#include <iostream>

using namespace std;

class base {
public:
   base()
   {
       cout << "Constructing base \n";
   }
   ~base()
   {
       cout << "Destructing base \n";
   }
};

class derived : public base {
public:
   derived()
   {
       cout << "Constructing derived \n";
   }
   ~derived()
   {
       cout << "Destructing derived \n";
   }
};

int main(void)
{
   derived* d = new derived();
   base* b = d;
   delete b;
```

```
    getchar();
    return 0;
}
```

23. What is the output of the below program?

```
#include <iostream>
using namespace std;
class BaseA
{
public:
    BaseA()
    {
        cout << "BaseA constructor called" << endl;
    }
};
class BaseB
{
public:
    BaseB()
    {
        cout << "BaseB constructor called" << endl;
    }
};
class Derived: public BaseA, public BaseB
{
public:
    Derived()
    {
        cout << "Derived's constructor called" << endl;
    }
};
int main()
{
    Derived d;
    return 0;
}
```

Output:

```
BaseA constructor called
BaseB constructor called
Derived’s constructor called
```

Explanation: When a class inherits from multiple classes, constructors of base classes are called in the same order as they are specified in inheritance.

PART-C 12 marks:

1. Define UML Sequence diagram with its notations. Draw the sequence diagram for Online hotel management
2. Explain UML collaboration diagram and its notations. Draw the collaboration diagram for online flight ticket reservation
3. Draw an sequence and collaboration diagram for online railway reservation system
4. Draw an interaction diagram for online banking system
5. Define interaction diagram. Draw an interaction diagram for online hospital management
6. Explain constructors and its types. Write a program for multiple constructors.
7. Write a C++ program for Constructor overloading (parameterized, default and copy).
8. Define interaction diagram. Draw the interaction diagram for ATM amount withdrawal
9. Define interaction diagram. Draw the interaction diagram for amount withdrawal
10. Explain in detail about operator overloading and its types with example
11. Explain the constructors types with example
12. Write a program for constructor overloading

UNIT-3
PART-A

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
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) firend
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

PART-B

1. What is Inheritance? List out the Benefits of Inheritance
2. Explain in detail about Virtual Function
3. What is pure virtual function?
4. What is abstract class?
5. What are advanced functions?
6. What is statechart diagram? Give its benefits
7. What is Activity diagram? Give its benefits
8. What is single inheritance and multiple inheritance?
9. Explain Hybrid Inheritance with example?
10. Explain the types of inheritance?
11. Explain the notations of Activity Diagram.
12. Explain the notations of Statechart Diagram
13. Draw UML State chart Diagram for C
14. Draw UML Activity Diagram for Patient registration system
15. Draw UML State chart Diagram for Order Management system
16. 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 (Justify the Answer)

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;
    }
```

Ans: 295 (Justify the Answer)

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;
}
```

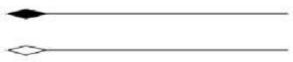
Ans: Error in statement: inline func1(2.2); (Justify the Answer)

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;
            }
```

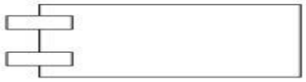
Output: n (Justify the Answer)

26. 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

27. 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

28. 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

29. Which are the situations where inline function is not working?

Some situations where inline function may not work -

For function returning values, if a loop, a switch or goto exists.
for function not returning values, if a return statement exists.
if functions contains static variables.
if inline functions are recursive.
if function contains large code.
if function is virtual as binding takes place at compile time.

30. How the inline functions can speed up the processing
Inline functions can speed up processing in 3 ways -

- Eliminating the overhead associated with the function call instruction.
- Eliminating the overhead associated with the pushing and popping parameters.
- Allowing the compiler to optimize the code outside the function call more efficiently.

PART-C

1. Explain Single Inheritance with example
2. Explain Pure Virtual Function with example program

3. Draw UML state chart and Activity Diagram for ATM processing
4. Illustrate UML state chart and Activity Diagram for Patient registration system
5. Elaborate ad draw UML state chart and Activity Diagram for Library Management System
6. Illustrate UML Interaction Diagram for E-Commerce registration system
7. Describe any two types of Inheritance with example program
8. Explain Multiple Inheritance with example program
9. Explain Multilevel Inheritance with example program
10. Discuss Abstract class and Interface with suitable Examples
11. What is Virtual function? Explain with examples
12. Explain advanced friend function and friend class with example
13. Explain Hierarchical Inheritance with example?
14. **What is the output of this program?**

```
#include <iostream>
using namespace std;
class Test {
protected:
    int width, height;

public:
    void set_values(int a, int b)
    {
        width = a;
        height = b;
    }
    virtual int area(void) = 0;
};
class r : public Test {
public:
    int area(void)
    {
        return (width * height);
    }
};
class t : public Test {
public:
    int area(void)
    {
        return (width * height / 2);
    }
};
```

```
int main()
{
   r rect;
   t trgl;
   Test* ppoly1 = ▭
   Test* ppoly2 = &trgl;
   ppoly1->set_values(4, 5);
   ppoly2->set_values(4, 5);
   cout << ppoly1->area();
   cout << ppoly2->area();
   return 0;
}
```

```
Output ; 2010
```

Explanation: In this program, we are calculating the area of rectangle and triangle by using abstract class.

15. What is the output of this program?

```
#include <iostream>
using namespace std;
class Base {
public:
   virtual void print() const = 0;
};
class DerivedOne : virtual public Base {
public:
   void print() const
   {
      cout << "1";
   }
};
class DerivedTwo : virtual public Base {
public:
   void print() const
   {
      cout << "2";
   }
};
class Multiple : public DerivedOne, DerivedTwo {
public:
   void print() const
   {
      DerivedTwo::print();
   }
};
int main()
{
   Multiple both;
   DerivedOne one;
   DerivedTwo two;
   Base* array[3];
   array[0] = &both;
   array[1] = &one;
   array[2] = &two;
   for (int i = 0; i < 3; i++)
      array[i]->print();
   return 0;
}
```

Output 212

Explanation: In this program, We are executing these based on the condition given in array. So it is printing as 212.

16. What is the output of this program?

```
#include <iostream>
using namespace std;
class sample {
public:
    virtual void example() = 0;
};
class Ex1 : public sample {
public:
    void example()
    {
        cout << "GeeksForGeeks";
    }
};
class Ex2 : public sample {
public:
    void example()
    {
        cout << " is awesome";
    }
};
int main()
{
    sample* arra[2];
    Ex1 e1;
    Ex2 e2;
    arra[0] = &e1;
    arra[1] = &e2;
    arra[0]->example();
```

```
   arra[1]->example();
}
```

Output: GeeksForGeeks is awesome

Explanation: In this program, We are combining the two statements from two classes and printing it by using abstract class.