IMAYAM ARTS & SCIENCE COLLEGE
KANNANUR-THURAIYUR-621 206.

DEPARTMENT OF COMPUTER SCIENCE AND APPLICATIONS

QUESTION BANK

CLASS	:	**B.SC .,COMPUTER SCIENCE & BCA**
YEAR	:	**I**
SEMESTER	:	**II**
TITLE OF THE PAPER	:	**PROGRAMMING IN C++**
SUBJECT CODE	:	**16SCCCS2 & 16SCCCA2**
PREPARED BY	:	**Ms. K. AKILA MCA., M.Phil.**

ONE MARK

UNIT – I BASIC CONCEPTS OF OBJECT

1. C++ is which type of language.

A High Level
B Low Level
C Middle Level
D None of the above

Ans: C. Middle Level

2. Class is a collection of

A a) Data member and member function
B Methods
C objects and functions
D data member and data function

Ans: A. Data member and member function

3. Objects are the basic run time

A Class
B Objects
C Entity
D None

Ans: C. Entity

4. Inheritance provides the idea of

A a) Reusability
B Independently
C Modularity
D Both a & b

Ans: A.Reusability

5. Dynamic binding is also known as

A Quick Binding
B Late Binding
C Dynamic Linking
D None

Ans: B. Late Binding

6. All variables must be declared before they are used in

A Executable Statement
B Running Statement
C Both A & B
D None of the above

Ans: A. Executable Statement

7. A function call passes arguments by

A Arguments
B Function
C Value
D None

Ans: C. Value

8. Loop structure is also known as

A Straight line
B Iteration
C Repetition
D Both B & C

Ans: D.Both B & C

9. Constant expressions consist of only

A Float values
B Rounded Values
C Constant Values
D None

Ans: C. Constant Values

10. C++ is

A Top down approach
B Bottom up approach
C Both A & B
D None

Ans: B. Bottom up approach

ONE MARK
UNIT – II CLASSES AND OBJECTS

1. Structure is a

A User defined data type
B Built in data type
C Both A & B
D None

Ans: A. User defined data type

2. Defining member function in

A One place
B Two place
C Three place
D None

Ans: B. Two place

3. A static member function can be called using the

A Class name
B Object name
C Static name
D None

Ans: A.Class name

4. A class member pointer can be declared using the following operator.

A ::
B *
C ::*
D *::

Ans: C. ::*

5. A static member function can have access to only other static members declared in the

A Same class
B Different class
C Both A & B
D None

Ans: A.Same class

6. A constructor has no arguments are callled

A Copy constructor
B Default constructor
C Both A & B
D None

Ans: B. Default constructor

7. Destructor has the same name as constructor and is preceded by

A +
B !
C ~
D √

Ans: C. ~

8. Overloaded operators cannot be

A Overridden
B Overloaded
C Overriding
D None

Ans: C. Overriding

9. One base class with more than one derived class is known as

A Single Inheritance
B Multiple Inheritance
C Multilevel Inheritance
D None

Ans: C. Multiple Inheritance

10. Pointer is a

A Derived data type
B Primary data type
C User defined data type
D None

Ans: A. Derived data type

ONE MARK
UNIT – III INHERITANCE

1. The old class is known as

A Base class B Parent class
C Both A and B D A only

Ans: C. Both A and B

2. The new class is known as

A Derived class B Child class
C Both A and B D B only

Ans: C. Both A and B

3. One base class with one derived class is called as

A Multiple inheritance B Single inheritance
C Multilevel inheritance D None

Ans: B. Single inheritance

4. When a base class is privately inherited by a derived class, public members of the base class become

A Public member B Private member
C protected member D none

Ans: B. Private member

5. A pointer can point to an object created by a

A Object B Class
C Both A And B D None

Ans: B. Class

6. this is a

A a)Address B Pointer
C Variable D Data

Ans: B. Pointer

7. The virtual function cannot be a

A Member B Variable
C Static member D None

Ans: C. Static member

8. The source stream that provides data to the program is called

A Input stream B Output stream
C Both D None

Ans: A. Input stream

9. The destination stream that receives output from the program is

A Input steam B Output stream
C Both A and B D None

Ans: B. Output stream

10. The classes istreamand ostreamdefine two member functions get() and Put() respectively to handle which input and output Operations.

A Single character B String
C Both A and B D None

Ans: A. Single character

ONE MARK

UNIT – IV MANAGING CONSOLE I/O OPERATIONS

1. Generic pointers can be declared with

A void B auto

C asm D None of the above

Ans: A.void

2. In which stream class is to only write on files?

A ofstream B ifstream

C fstream D iostream

Ans: A. ofstream

3. Which value is placed in the base class?

A Derived values B Default values

C Both default and derived D None

Ans: B.Default values

4. How many minimum numbers of functions need to be presented in C++?

A 0 B 1

C **2** D 3

Ans: B.1

5. Which of the following methods can be used to open a file in file handling?

A Using open() B Constructor method

C Both a & b D None

Ans: C.Both A & B

6. Which keyword is used to handle the exception?

A Trye B Throw

C Catch D None of the above

Ans: C. Catch

7. Which is used to throw a exception?

A Try B Throw

C Catch D None of the above

Ans: B. Throw

8. Which exception is thrown by dynamic_cast?

A bad_cast B bad_typeid

C bad_exception D bad_alloc

Ans: A.bad_cast

9. Return type of uncaught_exception() is

A int B bool

C cha D double

Ans: B.bool

10. Which type of program is recommended to include in try block?

A Static memory allocation B Dynamic memory allocation

C Cost reference D Pointer

Ans: B.Dynamic memory allocation

ONE MARK
UNIT – V STANDARD TEMPLATE LIBRARY

1. A container class is a class whose instances are

A Containers B functions
C string D none of the above

Ans: A. Containers

2. In how many categories containers are divided?

A 1 B 2
C 3 D 4

Ans: D.4

3. How many sequence containers provided by c++?

A 2 B 3
C 4 D 5

Ans: C.4

4. Which header fine is used for operators?

A <iterator> B <algorithm>
C <iter> D <loopiter>

Ans: A. <iterator>

5. sequence containers has

A 2 types B 3 types
C 4 types D None

Ans: B.3 types

6. coding refers to the translation of the design into

A machine readable form B design form
C coding form D None

Ans: A. machine readable form

7. How many steps bject oriented analysis have?

A 3 B 4
C 5 D 6

Ans: B. 5

8. Identification of objects has

A 3 Approaches B 2Approaches
C 4 Approaches D None

Ans: B. Approaches

9. Which type of diagram can be used to enlist the information.

A IFD B DFD
C ER D None

Ans: A. IFD

10. .In object oriented analysis the entities are called

A class B object
C method D function

Ans: B. Object

TWO MARKS
UNIT – I BASIC CONCEPTS OF OBJECT

1. Define Polymorphism.

Ans: Polymorphism is a Greek term. That means the ability to take more than one form This is one of the important OOPs concept

2. Define object.

Ans: Objects are the basic run time entities in an object oriented system. They may represent a person, a place, a bank account, a table of item that the program has to handle.

3. List out the application of OOPs.

Ans:
- Real-time system
- Simulation and modeling
- Object-oriented databases
- Hypertext, hypermedia and expert text

4. Rules for identifiers.

Ans:
- Only alphabetic characters, digits and underscores are permitted.
- The name can't start with a digit.
- Uppercase and lowercase letters are distinct.
- A declared keyword cannot be used as a variable name

5. Define tokens.

Ans: The smallest individual units in a program are known as tokens.
C++ has the following tokens:
- Keywords
- Identifiers
- Constants
- Strings
- Operators

6. Define encapsulation.

Ans: The wrapping up of data and functions into a single unit is known as encapsulation. Data encapsulation is the most striking feature of a class.

7. Define inheritance.

Ans: Inheritance is the process to acquire the properties of objects of one class to another class.

8. Uses of void.

Ans: To specify the return type of a function when it is not returning any value.
To indicate an empty argument list to a function.

9. Define class.

Ans: A class is a collection of objects of similar type. Classes are user defined data types and behave like built in types of programming language.
Mango, apple and orange are members of the class fruit.

10. What is dynamic binding?

Ans: Binding refers to linking the procedure call to the code to be executed in response to the call. Dynamic binding also known as late binding.

TWO MARKS
UNIT – II CLASSES AND OBJECTS

1. Define constructor.

Ans:

A constructor is a special member function whose task is to initialize the objects of its class. It is a special because it's name is the same as the class name. A constructor is invoked whenever an object of its associated class is created.

Syntax:

Class constructor name

Ex:

Class integer

2. How to specifying a class?

Ans:

A class is a way to bind the data and its associated functions together. It allows the data and functions to be hidden. When defining a class, we are creating a new abstract data type that can be treated like any other built in data type. A class specification has two parts:

1. Class declaration
2. Class function definitions

Syntax:

```
Class class name
{
 Private:
      Variable declaration;
       Function declarations;
  Public:
       Variable declarations;
        Function declarations;
};
```

3. Define arrays of object.

Ans:

An array can be of any data type including struct. We can also have array of variables that are of these type class. Such variables are called arrays of objects.

4. How object as function arguments?

An object may be used as a function argument. This can be done in two ways.

Ans: A copy of the entire object is passed to the function.

Only the address of the object is transferred to the function.

5. List out some characteristics of constructor.

They should be declared in the public section.

They are invoked automatically when the objects are created.

Ans: They do not have return types, not even void and therefore and therefore cannot return values.

We cannot refer to their address.

6. Define destructed.

Ans: A constructor is used to destroy the objects that have been created bus constructor. Like a constructor, the destructor is a member function whose name is the same as the class name but is preceded by a tilde.

```
~integer(){}
```

7. What are the operators does not overloaded in CPP?

Ans: Class member access operator, Scope resolution operator , Size operator and Conditional operator

8. List out the rules for overloading operators.

Only existing operators can be overloaded. New operators cannot be created.

The overloaded operator must have at least one operand that is of user-defined type.

Ans: We cannot change the basic meaning of an operator. That is to say, we cannot redefine the plus operator to subtract one value from the other.

There are some operators that cannot be overloaded.

9. What are the ways to pass the initial values as arguments to the constructor?

Ans: By calling the constructor explicitly

By calling the constructor implicitly

10. What are the conditions should satisfy the casting operator?

It must be a class member.

Ans: It must not specify a return type.

It must not have any arguments.

TWO MARKS

UNIT – III INHERITANCE

1. List out the types of inheritance.

Ans: Single inheritance, multiple inheritances, Hierarchical inheritance, Multi level inheritance and Hybrid inheritance.

2. Define single inheritance.

Ans: A derived class with only one base class is called single inheritance.

3. Define multiple inheritances.

Ans: One derived class with more than one base class is known as multiple inheritances.

4. Define hierarchical inheritance.

Ans: One base class with more than one derived class is known as hierarchical inheritance.

5. Define multi level inheritance.

Ans: The mechanism of deriving a class from another derived class is known as multilevel inheritance.

6. Define hybrid inheritance.

Ans: It is the combination of multiple and multilevel inheritance.

7. How to define a derived class.

Ans: A derived class can be defined by specifying its relationship with the base class in addition to its details.

Syntax: Class derived class name : visibility mode base class name { };

8. How to declare and declare and initializing pointers.

Ans: A pointer is a derived data type that refers to another data variable by storing the variables rather than data. The declaration of a pointer variable takes the following form.

Syntax: data type *pointer variable;

9. Define this pointer.

Ans: A unique keyword called this to represent an object that invokes a member function. This is a pointer that points to the object for which this function was called. The pointer this acts as an implicit argument to all the member functions. **This->a=123;**

10. What are the arithmetic Operations performed by pointer?

Ans: A pointer can be incremented or decremented. Any integer can be added to or subtracted from a pointer. One pointer can be subtracted from another.

TWO MARKS
UNIT – IV MANAGING CONSOLE I/O OPERATIONS

1. Define bug and it's types.

Ans: A bug is an error. There are two types: 1.Logical error 2.Syntatic error

2. Why logic and syntactic errors occurs?

Ans: The logic errors occur due to poor understanding of the problem and solution procedure. The syntactic error occur due to poor understanding of the language itself.

3. Define exception.

Ans: Exceptions are run time anomalies or unusual conditions that a program may encounter while executing. Anomalies include conditions such as division by zero, access to an array outside of its bounds.

4. What are the task performed in error handling mechanism?

Ans: Find the problem, Inform that an error has occurred, Receive the error information and Take corrective actions

5. How many ways we can use to open a file?

Ans: A file can be opened in two ways
1. Using the constructor function of the class.
2. Using the member function open() of the class.

6. List out the file stream classes.

Ans: 1. .seekg() 2.seekp() 3.tellg() 4.tellp()

7. Define template.

Ans: A template can be used to create a family of classes or functions. Templates allow us to define generic classes. Generic programming is an approach where generic types are used as parameters in algorithms.

8. Define parameterized classes.

Ans: A template is defined with a parameter that would be replaced by a specified data type at the time of actual use of the class or functions.
The templates are sometimes called parameterized classes or functions.

9. Define function template.

Ans: We can also define function templates that could be used to create a family of functions with different argument types. The general format for function template is:

```
template<class T>
returntype function name (arguments of type T)
{       }
```

10. Define class template.

Ans: Template allow us to define generic classes. It is a simple process to create a generic class using a template with an anonymous type. General form:

```
template<class T>
class classname
{           }
```

TWO MARKS

UNIT – V STANDARD TEMPLATE LIBRARY

1. Define STL and its components.

Ans: The collection of generic classes and functions is called the Standard Template Library. The components are Containers, Algorithms and iterators.

2. Define container.

Ans: Containers an object that actually stores data. It is a way data is organized in memory. The STL containers are implemented by template classes and therefore can be easily customized to hold different types of data.

3. Define Algorithm.

Ans: An algorithm is a procedure that is used to process the data contained in the containers. The STL includes many different kinds of algorithms to provide support to tasks. Algorithms are implemented by template functions.

4. Define operator.

Ans: An operator is an object that points to an element in a container. We can use operators to move through the contents of containers. Operators are handled just like pointers.

5. List out the types of containers.

Ans: Sequence containers, Associative containers and Derived containers.

6. List out the types of sequence containers.

Ans: Vector, List and Deque

7. List out the types of Associative containers.

Ans: Set, multi set, map, multi map

8. List out the types of derived containers.

Ans: 1.stack 2.queue 3.priority-queue

9. Expand OOP and POP.

Ans: OOP-Object Oriented Paradigm
POP-Procedure Oriented Paradigm

10. Define objects in entities.

Ans: Entities may represent physical objects such as equipment and people and abstract concepts such as data files and files and functions. In object oriented analysis the entities are called objects.

FIVE MARKS

UNIT – I BASIC CONCEPTS OF OBJECT

1. Explain about data types.
2. Write a short note on expressions.
3. Write about the benefits of OOPs.
4. Explain about function prototyping.
5. Write a note on call by reference.
6. Write a note on return by reference.
7. Explain about inline function with example.
8. Explain about function overloading.

UNIT – II CLASSES AND OBJECTS

1. Explain about classes with example.
2. Write a short note on defining member functions.
3. Write about arrays of objects.
4. Explain about friend function with example.
5. Explain about destructors.
6. Write about operator overloading.
7. Explain about overloading unary operator.
8. Explain about overloading binary operator.

UNIT – III INHERITANCE

1. Explain multiple inheritances with example.
2. Explain multilevel inheritance with example.
3. Explain about polymorphism.
4. Explain about virtual functions.
5. Write a note on rules for virtual functions.
6. Write a note on C++ stream classes.

UNIT – IV MANAGING CONSOLE I/O OPERATIONS

1. How to opening and closing a file explain.
2. Explain about file pointers and manipulations.
3. Explain about error handling during file Operations.
4. Explain about class template.
5. Explain about function template.
6. Explain about class template with multiple parameters.
7. Explain about function template with multiple parameters.
8. How to re growing exceptions.

UNIT – V STANDARD TEMPLATE LIBRARY

1. How to create string objects.
2. How to manipulate string objects.
3. Explain about relational Operations.
4. How to accessing characters in Strings.
5. How to compare and swapping a string.
6. Explain about procedure oriented development tools

TEN MARKS

UNIT – I BASIC CONCEPTS OF OBJECT

1. Briefly explain about the basic concepts of object oriented programming.
2. Write a brief note on operators.
3. Explain in detail about control structure.
4. Explain about expressions with example.

UNIT – II CLASSES AND OBJECTS

1. Explain in detail about constructors and their types.
2. Write a brief note on rules for operator overloading.

UNIT – III INHERITANCE

1. Write a brief note on unformatted I/O Operations.
2. Explain in detail about inheritance and it's types with example.
3. Briefly explain about formatted console I/O Operations.

UNIT – IV MANAGING CONSOLE I/O OPERATIONS

1. Explain in detail about exception handling with example.
2. Briefly explain about various fine Operations

UNIT – V STANDARD TEMPLATE LIBRARY

1. Briefly explain about STL components.
2. Explain in detail about procedure oriented paradigm
3. Write a brief note on object oriented paradigm.
4. Briefly explain about object oriented notations and graphs.
5. Explain about steps in object oriented analysis.
6. Explain about steps in object oriented design.