IDHAYA COLLEGE FOR WOMEN
KUMBAKONAM

SEMESTER	:	II
SUBJECT	:	PROGRAMMING IN C++
SUBJECT CODE	:	16SCCCA2
TOPIC	:	UNIT I - V (Two Marks)

FACULTY NAME

Mrs.V.KALAIVANI, M.Sc., M.Phil.,

ASSISTANT PROFESSOR,

DEPARTMENT OF COMPUTER APPLICATIONS,

IDHAYA COLLEGE FOR WOMEN,

KUMBAKONAM.

UNIT I

1**. What is procedure oriented programming?**

Procedure oriented programming is a style of programming where procedures are the main structural feature of programs and basic data types are the main form of representation of data.

2**.What are the draw backs of procedure oriented programming?**

Data move freely around the program and there fore vulnerable to changes caused by any function in the program and it does not model very well the real world problems.

3. What is object oriented programming?

Object oriented programming is a programming paradigm based on the concept of “objects”,which contain data in the form of fields and code in the form of procedures.

4.Write some applications of Oops?

Real time Systems,Simulation and Modeling,AI and expert Systems,Neural networks and parallel programming.

5.What are the Basic concepts of Object oriented programming?

- Objects
- Classes
- Data abstraction and encapsulation
- Inheritance
- Polymorphism
- Dynamicbinding
- Message passing

6.Define Object.

Objects are the basic runtime entities in an object oriented system.In other words Object is an entity that has state and behaviour.

7.Define Class.

A class in C++ is a user-defined type or data structure declared with keyword class that has data and functions (also called member variables and member functions) as its members whose access is governed by the three access specifiers private, protected or public. By default access to members of a C++ class is private.

8.Define Data Abstraction.

Abstraction refers to the act of representing essential features without including the background details or explanation.

9.Define Encapsulation.

The wrapping up of data and functions into a single unit is known as encapsulation.The data is not accessible to the outside world and only those functions which are wrapped in the class can access it.

10.What is the use of // symbol in c++?

The double slash comment is basically a single line comment.Comments start with a double slash symbol and terminate at the end of the line.A comment may start anywhere in the line,and whatever follows till the end of the line is ignored.

11.How the multiline comments will be specified in C++?

The comments /*,*/ are used to specify multiline comments.

Eg:/* This is an example program

to specify important features */

12.What is the use of the header file iostream?

The header file iostream should be included at the beginning of all programs that use input/output statements.The iostream contains the declarations for cout and cin.

13.What is the use of #include directive?

All C++ programs begin with a #include directive that includes the specified header file contents into the main program.

For eg: #include <iostream>

This directive causes the preprocessor to add the contents of the iostream file to the program.

14.What is the use of >> operator?

The operator >> is known as extraction or get from operator.It extracts the value from the keyboard and assigns it to the variable on its right.

Cin>> variable;

15.Define Token.

The smallest individual units in a program are known as tokens.C++ has the following tokens.

Keywords

Identifiers

Constants

Strings

Operators

16.Define Keyword.

The keywords are reserved identifiers and cannot be used as names for the program variables or other user defined program elements.

Eg: double,int,string,char,if,while,do

17.What is identifier?List the rules for naming the identifiers.

Identifiers refer to the names of variables,functions,arrays,classes created by the programmer.

Rules for naming the identifier:

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

18.Define Constant.

Constant refer to fixed values that do not change during the execution of a program.

19.How the datatypes in C++ classified?

In C++ the data types are classified as

User-defined type

Built-in type

Derived type

20.What are the basic data types in C++?

The built in datatypes in C++ are known as basic or fundamental data types.The basic data types are:int,float,char,double

21.List the User defined datatypes.

Structure,union,class,enumeration

22.List the derived data types.

Array,function,pointer,reference.

23.List the four types of storage classes.

Automatic,External,Static,Register

24.What is dynamic initialization of variable?

In C++,a variable can be initialized at run time using expressions at the place of declaration is known as dynamic initialization of variable.

Eg:

float area = 3.14 * r *r;

25.What is reference variable?

A reference variable provides an alias for a previously defined variable.A reference variable is created as follows:

data-type&reference-name = variable-name;

Eg:

float total = 100;

float &sum = total;

26.What is scope resolution operator?

The operator :: is known as Scope resolution operator.It takes the following form.

:: variable-name

This operator allows access to the global version of a variable.

27.What are Manipulators?

Manipulators are operators that are used to format the data display.The most commonly used manipulators are endl and setw.

28.What is the use of type cast operator?

C++ permits explicit type conversion of variables or expressions using the type cast operator.

type-name(expression)

Eg: average =sum/float(i);

29.What is expression and their types?

An expression is a combination of operators ,constants and variables arranged as per the rules of the language.It may also include function calls which return values.An expression may consist of one or more operands and zero or more operators to produce a value.

Types of Expressions:

- Constant expression
- Integral expression
- Float expression
- Pointer expression
- Relational expression
- Logical expression
- Bitwise expression

30.Define Implicit Conversion.

In C++ Wherever the data types are mixed in an expression,itperforms conversions automatically.This process is known as implicit or automatic conversion.

31.What are the basic control structures?

1.Sequence structure

2.Selection structure.

3.Loop structure .

32.What is function prototyping?

The function prototyping describes the function interface to the compiler by giving details such as the number and type of arguments and the type of return values.

The function prototyping is a declaration statement in the calling program and is of the following form.

```
type function-name (argument-list);
```

33.What is inline function?

An inline function is a function that is expanded in line when it is invoked.That is,the compiler replaces the function call with the corresponding function code.

Syntax:

```
inline function-header
{
function body
}
```

34.Define function overloading.

Function overloading means the same function name is used to create functions that perform a variety of different tasks.The function would perform different operations depending on the argument list in the function call.The correct function to be invoked is determined by checking the number and type of arguments but not on the function type.

UNIT – II

1.Define Constructor.

A constructor is a special member function whose task is to initialize the objects of its class.It is special because its name is the same as the class name.

The constructor is invoked whenever an object of its associated class is created.

A constructor is declared and defined as follows:

```
// class with a constructor
class integer
{
int m,n;
public:
  integer (void);              // constructor declared
    ………
    ……….
  };
integer::integer(void)         //constructor defined
m=n=0;
  }
```

2.Define Default Constructor.

A constructor that accepts no parameters is called default constructor.

If no constructor is defined,then the compiler supplies default constructor implicitly.

3.Define Parameterized Constructor.

The constructors that take arguments are called parameterized constructor.

It initializes the various data elements of different objects with different values when they are created.

Eg:

```
class integer
{
int m,n;
public:
 integer (int x,int y);              //parameterized constructor
     ………
       ……….
};
integer::integer(int x,int y)
```

```
m=x;n=y;

  }
```

4.What is dynamic initialization?

The initial value of an object may be provided during run time is called dynamic initialization.Dynamic initialization is possible by using overloaded constructors.

5.Define Copy Constructor.

A copy constructor is a constructor which is used to declare and initialize an object from another object.

Eg:

```
integer I2 (I1);
```

It defines the object I2 and at the same time initialize it to the values of I1.

6.Define Dynamic Constructor.

The constructor which enables the system to allocate memory to objects at runtime is called dynamic constructor.The memory is allocated with the help of new operator.

7.Define Destructor.

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

Eg:

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

8.Define Operator overloading.

The mechanism of giving special meanings to an operator is known as operator overloading.

All the c++ operators can be overloaded except the following.

- Class member access operator
- Scope resolution operator
- Sizeof operator
- Conditional operator

Syntax:

```
return type classname : : operator op ( arglist)

{

Function body // task defined

}
```

UNIT III

1.Define Inheritance.

The mechanism of deriving a new class from an old one is called inheritance.The old class is referred to as the base class and the new one is called the derived class or subclass.

2.What are the types of Inheritance?

Types of Inheritance:

- Single Inheritance
- Multiple Inheritance
- Multilevel Inheritance
- Hierarchical Inheritance
- Hybrid Inheritance

3.How to define a derived class?

A derived class can be defined by specifying its relationship with the base class in addition to its own detail.

Syntax:

```
Class  derived-classname : visibility-mode  base-classname

{

…………….. //

…………….//  members of derived class

……………//
};
```

The visibility mode may be either private or public.

4.What is Pointer?

A pointer is a derived data type that refers to another data variable by storing the variable's memory addresss rather than data.

Syntax:

```
data – type   *pointer – variable;
```

5.Define This pointer.

This is a pointer that points to the object for which this function was called.

This unique pointer is automatically passed to a member function when it is called.

The pointer this acts as an implicit argument to the member functions.

6.Define Virtual function.

When same function name is used in both the base and derived classes,the function in base class is declared as virtual using the keyword virtual preceding its normal declaration.

When a function is made virtual,C++ determines which function to use at runtime based on the type of object pointed to by the base pointer,rather than the type of the pointer.

7.Define Pure Virtual function.

A Pure Virtual function is a function declared in a base class that has no definition relative to the base class.

8.Define Abstract class.

A class containing a pure virtual function is called an Abstract class.

UNIT-IV

1.Define Stream.

A stream is a sequence of bytes.It acts either as a source from which the input data can be obtained or as a destination to which the output data can be sent.

2.What are the types of Streams?

Input Stream:

The source stream that provides data to the program is called the input stream

Output Stream:

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

3.Define Stream Classes.

The C++ I/O system contains a hierarchy of classes that are used to define various streams to deal with both the console and disk files.These classes are called stream classes.

4.Define Command Line Arguments

File names supplied as arguments to the main() function at the time of invoking the program .These arguments are known as command line arguments.

5.What is the use of get() function?

Reading a single character from the stream can be done by using the get( ) function.

Syntax:

get(variablename);

Eg:

char name;

get (name);

6.What is the use of put() function?

put() is a member of ostream class ,can be used to output a line of text,character by character.

Eg:

Cout.put('x');

7.What is the use of getline() function?

The getline() function reads a whole line of text that ends with a newline character.The function can be invoked by using the object cin as follows.

```
cin.getline(line,size);
```

8.What is file?

A file is a collection of related data stored in a particular area on the disk.

9.How to open a file using Constructor?

A constructor is used to initialize an object while it is being created.Here ,a filename is used to initialize the file stram object.It involves the following steps:

1.Create a filestream object to manage the stream using the appropriate class.
2.initialize the file object with the desired filename.

Eg:

```
ofstream  outfile(“results”);
```

10.What is template?

C++ supports a mechanism known as template to implement the concept of generic programming.Template allows us to generate a family of functions to handle different data types.

11.Define Template Class.

A class created from a class template is called a template class.

The general form of class template is :

```
template<class T>

class   classname

{

//………

//………

//……..

};
```

The syntax for defining an object of a template class is:

Classname<type> objectname (arglist);

12.What are the two most common types of bugs?

Logic Errors

Syntactic Errors

13.Define Exceptions.

Exceptions are run time anomalies or unusual conditions that program may encounter while executing.Anomalies might include conditions such as division by zero,access to an array outside of its bounds, or running out of memory or disk space.

14.What are the two types of Exceptions?

The two types of Exceptions are:

Synchronous exception

Asynchronous exception

15.Define ASynchronous exceptions

The errors that are caused by events beyond the control of the program are called ASynchronous exceptions

16.Define Synchronous exceptions.

The errors such as “out of range index” and “overflow” known as Synchronous exception.

17.What is Exception Handling?

Exception Handling is a mechanism which detects and report an “exceptional circumstance” so that appropriate action can be taken.This mechanism suggests a separate error handling code that performs the following tasks:

1.Find the problem(Hit the exception)

2.Inform that an error has occurred(Throw the exception)

3.Receive the error information(Catch the exception)

4.Take corrective actions(Handle the exception)

18.Define Exception Handling mechanism.

The Exception Handling mechanism is basically built upon three keywords namely try,throw and catch.

The try block consists of set of statements which may generate exception.

When an exception is detected,it is thrown using a throw statement in the try block.

The catch block catches the exception thrown by the throw statement in the try block and handles the error appropriately.

Syntax:

```
try
{
……..
……..      /* Block of statements which
……..           detects and throws an exception */
throw  exception;
……..
}
catch  (type arg)  // Catches exception
{
………….
…………….
……………… //Handles exception
…………….
}
```

UNIT-V

1.What is Standard Template Library?

The collection of generic classes and fuctions is called Standard Template Library.

2.What are the benefits of Standard Template Library?

Save considerable time and effort

Lead to high quality program

3.What are the components of STL?

Containers

Algorithms

Iterators

4.What is Container?

A container is 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.

5.What is Algorithm?

An algorithm is a procedure that is used to process data contained in the containers.The STL includes many different kinds of algorithms to provide support to tasks such as initializing,searching,copying,sorting and merging.

6.Define Iterator.

An Iterator is an object that points to an element in a container.Iterators connect algorithms with containers and play a keyrole in the manipulation of data stored in the containers.

7.List the three types of Container.

- Sequence Container
- Associative Container
- Derived Container

THANK YOU