[HALF YEARLY EXAMINATION KEY ANSWER -2024-2025]

STD: XI – COMPUTER SCIENCE – DR SURESH MATRIC HSS – RAMANATHAPURAM (DIST)

I **(One word)**

1. D 2.A 3.C 4.D 5.C 6.D 7.B 8.D 9.D 10.C 11.B 12.B 13.D 14.D 15.C

II **(Two Marks) (Q.no 24 is compulsory)**

16

Primary memory	Secondary memory
❖ The Primary Memory is used to temporarily store the programs ❖ It is volatile in nature	❖ The Secondary memory is used to store the data permanently. ❖ It is non-volatile in nature
❖ Ex : RAM	❖ Ex: Hard Disk, CD or DVD

17

1) $\overline{A+B} = \overline{A} \cdot \overline{B}$ 2) $\overline{(A \cdot B)} = \overline{A} + \overline{B}$

18 (1) File access level (2) System level (3) Network level

19
❖ An invariant the loop body is known as a loop invariant.
❖ The property of the variables which remains unchanged by the execution of the loop body is called as loop invariant.

20 1.To indicate the function does not return a value 2.To declare a generic pointer

21 The declaration of a 2-D array is : data-type array_name[row-size][col-size];

22 A cookie is a small piece of data sent from a website and stored on the user's computer memory (Hard drive) by the user's web browser while the user is browsing internet.

23
❖ TSCII (Tamil Script Code for Information Interchange) is the first coding system to handle our Tamil language.
❖ This encoding scheme was registered in IANA (Internet Assigned Numbers Authority) unit of ICANN.

24 **Output:** 1) 178.2525 2) 178

III **(Three Marks) (Q.no 33 is compulsory)**

25 i)$1101010_2+101101_2$ = **10010111_2** ii) - $22_{10}+15_{10}$ = **-7_{10}** = **$(11111001)_2$**

26
❖ Case analysis statement generalizes it to multiple cases.
❖ Case analysis splits the problem into an exhaustive set of disjoint cases.
❖ For each case, the problem is solved independently.

Ex: If C1, C2 and C3 are conditions, and S1,S2,S3 and S4 are statements, a 4-case analysis statement has the form.

```
1. case C1
2.       S1
3. case C2
4.       S2
5. case C3
6.       S3
7. else
8.            S4
```

27

isuppr()	toupper()
❖ This function is used to check the given character is uppercase.	❖ This function is used to convert the given character into its uppercase.
❖ General form: isupper(char c);	❖ General form: toupper(char c);
❖ Ex: **int n=isupper('A');**	❖ Ex: **char c = toupper('k'); -- K**

28
❖ The overloaded function must differ in the number of its arguments or data types.
❖ The return type of over loaded functions are not considered for overloading same data type.
❖ The default arguments of overloaded functions are not considered as part of the parameter list in function over loading.

29

=	= =
❖ It is a assignment operator)	❖ It is a relational operator
❖ It is used to assign a value of variable or expression.	❖ It used to compare two values and the result will be either true or false.
❖ Ex: x = y (y value is assigned to x)	❖ Eg: x = = y (x value will be compared with y value)

30
❖ A structure without a name/tag is called anonymous structure.

Example:

```
struct
{
long rollno;
int age;
float weight;   } student;
```

- The student can be referred as reference name to the above structure and the elements can be accessed like student.rollno, student.age and student.weight.

31

1. Access applications on the computer (Ex: Word processing, Games, Spread sheets, Calculators]
2. Load any new program on the computer.
3. Manage hardware such as printers, scanners, mouse, digital cameras etc.,
4. File management activities [Ex: Creating, Modifying, Saving, Deleting files and folders]
5. Change computer settings. [Colour scheme, Screen savers of our monitor etc]

32

❖ There are two types of microprocessors based on their instruction sets.

1) Reduced Instruction Set Computers (RISC) :

❖ Example: Pentium IV, Intel P6, AMD K6 and K7.

2) Complex Instruction Set Computers (CISC) :

❖ Example: Intel 386 & 486, Pentium, Pentium II and III, and Motorola 68000.

33

```
#include<iostream>
using namespace std;
int main()
{
int n;
for(int i=1;i<=40,i+=3)
cout<<i<< endl;
getch ( );
}
```

Output:

1 4 7 10 13 16 19 22 25 28 31 34 37 40

IV **(Five Marks)**

34 a

Generation & Period	Main Component used	Merits/Demerits
First Generation 1940-1956	**Vacuum tubes**	❖ Big in size ❖ Consumed more power ❖ Malfunction due to overheat ❖ Machine Language was used
Second Generation 1956-1964	**Transistors**	❖ Smaller compared to First Generation ❖ Generated Less Heat ❖ Consumed less power compared to first generation ❖ Punched cards were used ❖ First operating system was developed – Batch Processing and Multiprogramming Operating System ❖ Assembly language was used.
Third Generation 1964-1971	**Integrated Circuits (IC)**	❖ Computers were smaller, ❖ faster and more reliable ❖ Consumed less power ❖ High Level Languages were used
Fourth Generation 1971-1980	**Microprocessor** Very Large Scale Integrated Circuits (VLSI)	❖ Smaller and Faster ❖ Microcomputer series such as IBM and APPLE were developed ❖ Portable Computers were introduced.
Fifth Generation 1980 – till date	Ultra Large Scale Integration(ULSI)	• Parallel Processing • Super conductors • Computers size was drastically reduced. • Can recognise Images and Graphics • Introduction of Artificial Intelligence and Expert Systems • Able to solve high complex problems including decision making • Logical reasoning
Sixth Generation In future	❖ Parallel and Distributed computing ❖ Computers have become smarter, faster and smaller ❖ Development of robotics ❖ Natural Language Processing ❖ Development of Voice Recognition Software	

Or

❖ The Distributed Operating System is used to access shared data and files that reside in any machine around the world using internet / intranet.

❖ The user can handle the data from different locations.

❖ The users can access as if it is available on their own computer.

Advantages :

- A user at one location can make use of all the resources available at another location over the network.
- Many computer resources can be added easily in the network
- Improves the interaction with the customers and clients.
- Reduces the load on the host computer.

35 a

1. Syntax Error:

- Syntax is a set of grammatical rules to construct a program.
- Every programming language has unique rules for constructing the source code.
- Syntax errors occur when grammatical rules of C++ are violated.

Example: if you type as follows, C++ will throw an error.

cout << "Welcome to Programming in C++"

2.Semantic Error:

- A Program has not produced expected result even though the program is grammatically correct.
- It may be happened by wrong use of variable / operator / order of execution etc.
- This means, program is grammatically correct, but it contains some logical error.
- So, Semantic error is also called as "Logic Error".

3.Run-time error:

- A run time error occurs during the execution of a program.
- It occurs because of some illegal operation that takes place.

For example, if a program tries to open a file which does not exist, it results in a run-time error.

OR

- Call by value method copies the value of an actual parameter into the formal parameter of the function.
- In this case, changes made to formal parameter within the function will have no effect on the actual parameter.

Example Program:

```
#include<iostream>
using namespace std;
void display(int x)
{
x=x*x;
cout<<"\n\nThe Value inside display function (x*x):"<<x;
}
int main()
{
int a;
cout<<"\nExample : Function call by value:";
cout<<"\n\nEnter the Value for A :";
cin>>a;
display(a);
cout<<"\n\nThe Value inside main function "<<a;
return(0);
}
```

Output

```
Enter the Value for A : 5
The Value inside display function (a * a) : 25
The Value inside main function: 5
```

36 a

Procedural Programming.	Object Oriented Programming
❖ It deals with algorithms	❖ It deals with data
❖ Programs are divided into functions	❖ Programs are divided into objects
❖ Less secure	❖ More secure
❖ It is top down approach	❖ It is bottom down approach
❖ All data items are global.	❖ Data abstraction is introduced.
❖ Emphasizes on algorithm.	❖ Emphasizes on data rather than algorithm.
❖ Overloading is not possible	❖ Overloading is possible
❖ Implement programs in the form of sub programs.	❖ Implement programs using classes and objects.
Ex: C,VB,COBOL, FORTRAN	Ex: C++, JAVA, VB.NET,PYTHON

OR

Constructor	**Destructor**
❖ The name of the constructor must be same as that of the class.	❖ The destructor has the same name as that class prefixed by the tilde character '~'.
❖ No return type can be specified for constructor.	❖ It has no return type
❖ A constructor can have parameter list.	❖ The destructor cannot have arguments.
❖ The constructor function can be overloaded.	❖ Destructors cannot be overloaded.
❖ They cannot be inherited but a derived class can call the base class constructor.	❖ They cannot be inherited

	❖ The compiler generates a constructor, in the absence of a user defined constructor.	❖ In the absence of user defined destructor, it is generated by the compiler.
	❖ The constructor is executed automatically when the object is created.	❖ The destructor is executed automatically when the control reaches the end of class scope to destroy the object.
	❖ Allocated memory space for the object	❖ Destroy the object

37a [CHAPTER-3 BOOK BACK Q.NO: 3 TWO MARK QUESTION] [8,16,32,64 –bit microprocessor]

OR

❖ There are different types of inheritance viz., Single Inheritance, Multiple inheritance, Multilevel inheritance, hybrid inheritance and hierarchical inheritance.

1. Single Inheritance :

❖ When a derived class inherits only from one base class, it is known as single inheritance.

2. Multiple Inheritance:

❖ When a derived class inherits from multiple base classes it is known as multiple inheritance

3. Hierarchical inheritance:

❖ When more than one derived classes are created from a single base class, it is known as Hierarchical inheritance.

4. Multilevel Inheritance:

❖ The transitive nature of inheritance is reflected by this form of inheritance.

❖ When a class is derived from a class which is a derived class – then it is referred to as multilevel inheritance.

5. Hybrid inheritance:

❖ When there is a combination of more than one type of inheritance, it is known as hybrid inheritance.

❖ Hence, it may be a combination of Multilevel and Multiple inheritance or Hierarchical and Multilevel inheritance or Hierarchical, Multilevel and Multiple inheritance.

38 a

Crime	Function
Cyber Terrorism	Hacking, threats, and blackmailing towards a business or a person.
Cyber stalking	Harassing through online.
Malware	Malicious programs that can perform a variety of functions including stealing, encrypting or deleting sensitive data, altering or hijacking core computing functions and monitoring user's computer activity without their permission.
Denial of service attack	Overloading a system with fake requests so that it cannot serve normal legitimate requests.
Fraud	Manipulating data, for example changing the banking records to transfer money to an unauthorized account.
Harvesting	A person or program collects login and password information from a legitimate user to illegally gain access to others' account(s).
Identity theft	It is a crime where the criminals impersonate individuals, usually for financial gain.
Intellectual propertytheft	Stealing practical or conceptual information developed by another person or company.
Salami slicing	Stealing tiny amounts of money from each transaction.
Scam	Tricking people into believing something that is not true.
Spam	Distribute unwanted e-mail to a large number of internet users.
Spoofing	It is a malicious practice in which communication is send from unknown source disguised as a source known to the receiver.

OR

L.n	Given code	Correct Code
1.	%include(iostream.h)	#include<iostream.h>
3.	Class A( )	Class A
5.	public;	public:
6.	int a1,a2:a3;	int a1,a2,a3;
7.	Void getdata[ ]	void getdata( )
10.	a2=13; a3=13;	a2=14; a3=13; (In order to get the given output)
12.	}	};
13.	Class B:: public A()	class B:: public A
15.	PUBLIC	public:
16.	voidfunc()	void func()
18.	int b1:b2:b3;	int b1,b2,b3;
19.	A::getdata[];	A::getdata( );
22.	a3=a3;	b3=a3;
23.	cout<<b1<<'\t'<<b2<<'t\'<<b3;	cout<<b1<<'\n'<<b2<<'\n'<<b3;
24	}	};
28	der1:func();	der:func();