



Atal Bihari Vajpayee
Indian Institute of Information Technology
and Management (ABV-IIITM), Gwalior
(An Institute of National Importance, Ministry of Education, Government of India)

MAJOR EXAMINATION-2025

Course Code: CS/IT-103

Course Name: Object Oriented Programming in C++

Program & Sem: B.Tech (IMT/IMG), 2nd Semester

Date: 30-04-2025 (Wed.)

Max Marks: 45

Instruction:

(i) Read all questions carefully and answer accordingly.

Part A

Answer all the Questions.

Each question carries *one* marks.

(15 Questions x 1 Mark = 15 Marks)

Q1. Which feature of OOP is indicated by the following code?

```
abstract class student
{ public: int marks;   calc_grade();   };
class topper:public student
{ public:   calc_grade() { return 10; } };
class average:public student
{ public:   calc_grade() { return 20; } };
class failed { int marks;;}
```

- a. Encapsulation and Inheritance b. Abstraction and Polymorphism ☒ c. Both a and b d. Polymorphism.

Q2. Choose the correct statement about the static member function:

- a. Object instantiation is not mandatory for calling static member functions. ☒ b. can't access this pointer.
c. can only access static member data and functions in a class. ☒ d. All are correct.

Q3. What is the output of this program?

```
#include<iostream>
using namespace std;
int main()
{ int a=10, b=10, c;
  c= a- - + ++b; - - c; c- - ;
  cout<<a<<" "<<b<<" "<<c; return 0;}
```

- a. 10 11 21 b. 9 12 21 c. 10 12 20 ☒ d. 9 11 19

Q4. A class hierarchy

- ☒ a. describes "is a kind of" relationships. b. shows the same relationships as an organization chart.
c. describes "has a" relationships. d. shows the same relationship as a family tree.

Q5. What is a pure virtual function in C++?

- a. A virtual function defined in a base class ☒ b. A virtual function declared in a base class
c. Any function in a class d. All of the above

Q6. Which among the statement/statements are correct about virtual functions:

- ☒ a. Implemented for late binding. b. Virtual functions can be static members.
c. Virtual functions should not be defined if they are not used. d. Virtual Destructors and Constructors are possible.

Q7. Which of the following is not a type of Constructor?

- a. Conversion. b. Copy ☒ c. Virtual d. None.

Q8. While overloading binary operators using member function, how many argument/s are required?

- a. Zero ☒ b. One c. Two d. Three

<pre> PTR = x; } void Sample::print() { } cout << *PTR - EOF << " "; } int main() { Sample S; S.set(10); S.print(); return 0; } </pre>	<pre> Derived() { cout<<"Constructing Derived \n"; } ~Derived() { cout<<"Destructing Derived \n"; } }; int main() { Derived *d = new Derived(); Base *b = d; delete b; return 0; } </pre>
<p>c.</p> <pre> #include <iostream> using namespace std; class education { char name[10]; public: int disp() {cout<<"Its our education system";return 0;} }; class school: public education { public: void disp() {cout<<"Its school education system";} }; int main() { school s; s.disp(); return 0; } </pre> <p><i>It's our education System</i></p>	<p>d.</p> <pre> #include <iostream> #include <string.h> using namespace std; class A { public: virtual void fun() = 0; }; class B : public A { public: void fun() { cout << "B::fun() called" << endl; } }; class C : public A { public: void fun() { cout << "C::fun() called" << endl; } }; int main() { A* p; B ob2; C ob3; p = &ob2; p->fun(); p = &ob3; p->fun(); return 0;} </pre> <p><i>B::fun() called C::fun() called</i></p>
<p>e.</p> <pre> #include <iostream> #include <math.h> using namespace std; namespace NamespaceOuter { int radius = 10; namespace NamespaceInner { int* ptr = &NamespaceOuter::radius; } namespace NamespaceFun { float calcuteArea() { float AreaOfCircle = 0.0F; AreaOfCircle = 3.14 * pow(*NamespaceOuter::NamespaceInner::ptr, 2); return AreaOfCircle; } } } int main() { cout << "Area Of Circle: " << NamespaceOuter::NamespaceFun::calcuteArea() << endl; return 0;} </pre> <p><i>Area of Circle : 314</i></p>	<p>f.</p> <pre> #include <iostream> using namespace std; class Sample1 { int A, B; friend class Sample2; }; class Sample2 { int X, Y; public: Sample2() { X = 5; Y = 5; } void fun() { Sample1 S; S.A = 10 * X; S.B = 20 * Y; cout << S.A << " " << S.B << endl; } }; int main() { Sample2 S; S.fun(); return 0;} </pre> <p><i>50 100</i></p>
<p>g.</p> <pre> #include<iostream> using namespace std; class demo {static int cnt; int data; public: demo(int d) { data=d; cnt++; } ~demo() { cnt--; } void show() {cout<<data<<cnt;} static void fun(demo & ob) {demo temp (ob.data); </pre> <p><i>{ 8 8 0 8 }</i></p>	<p>h.</p> <pre> #include <iostream> using namespace std; int main() { int A = 10, B = 5, C = 2; switch (A * ++B + C - 8) { default: <i>India Nepal Australia</i> cout << "India"; case 0x09: cout << "Nepal"; case 0x0A: cout << "Australia "; break; case 0x0B: </pre>

<pre>temp.show(); ob.show();} }; int demo::cnt; int main() { demo ob1(20); ob1.show(); demo::fun(ob1); ob1.show(); ob2.show(); return 0; }</pre>	<pre>cout << "USA "; break; case 0x0C: cout << "England "; break; } return 0;}</pre>
<p>i.</p> <pre>#include<iostream> using namespace std; class construct { int p,q; construct(int x, int y) { p=x; q=y; } public: void Display() {cout<<p<<"\n"<<q<<"\n";} }; int main() { construct item1(10,20), item2=construct(30,40); item1.Display(); item2.Display(); return 0;}</pre> <p style="color: red; text-align: center;">{ 8808 }</p>	<p>j.</p> <pre>#include<stdlib.h> #include<stdio.h> #include<iostream> using namespace std; class Test { int x; void* operator new(size_t size); void operator delete(void*); Test(int i) { x = i; cout << "Constructor called \n"; } ~Test() { cout << "Destructor called \n"; } }; void* Test::operator new(size_t size) { void *storage = malloc(size); cout << "new called \n"; return storage; } void Test::operator delete(void *p) { cout<<"delete called \n"; free(p); } int main() { Test *m = new Test(5); delete m; return 0; }</pre> <p style="color: red; text-align: center;">{ 8808 }</p>

Part C

Answer all the Questions.

Each question carries five marks.

(2 Questions x 5 marks = 10 Marks)

- Q17. a. Explain the importance of Constructors and Destructors in Object Oriented Programming. Discuss various types of Constructors in OOPs? (3 marks)
- b. What is polymorphism and its types. (2 marks)
- Q18. a. Write a C++ program using classes and using constructors to print Floyd's triangle upto 4 rows? (2 marks)
- b. We want to store the information of different vehicles. Create a class named Vehicle with two data member named mileage and price. Create its two subclasses: (3 marks)
- Car with data members to store ownership cost, warranty (by years), seating capacity and fuel type (diesel or petrol).
 - Bike with data members to store the number of cylinders, number of gears, cooling type(air, liquid or oil), wheel type(alloys or spokes) and fuel tank size(in inches)
- Make another two subclasses: Audi and Ford of Car, each having a data member to store the model type.
- Next, make two subclasses Bajaj and TVS of Bike, each having a data member to store the make-type.
- Now, store and print the information of an Audi and a Ford car (i.e. model type, ownership cost, warranty, seating capacity, fuel type, mileage and price.) Do the same for a Bajaj and a TVS bike.