MIT SCHOOL OF ENGINEERING

Rajbaug, Loni-Kalbhor, Pune

MIT-ADT UNIVERSITY PUNE, INDIA

Name: Sourav Toshniwal Class: FY-CSE Batch: A2

Subject: OOP Roll No.: 2213047 Exp. No.: 1

Name of the Experiment: Basics of Object-oriented Programming.

Performed on:

Submitted on:

Marks	Teacher's Signature with date
10	

MCQs:

1) Which of the following is known as the insertion operator in C++?

✓ a) <<

b) >>

c) ~

d) ^^

2) In C++, the cin and cout are?

✓ a) Predefined objects

b) Operators

c) Predefined pointers

d) Keywords

3) which operator is used for accessing a member of a namespace?

Incomplete for Theory ☐ Diagrams ☐ Observation Table ☐

Calculations ☐ Graphs ☐ Results ☐ Conclusion ☐

Understanding Through Q/A ☐ Late Submission ☐ Neatness ☐

Teacher's Signature

a) asterisk (*)
b) Scope (::)
c) Arrow (->)
d) Dot (.)

4) Predict the output

```
Class Test {
    int x;
};
int main()
{
 Test t;
 cout << t.x;
 return 0;
}
```

a) 0
b) Garbage value
c) Compilation error
d) None of the above.

5) How many specifiers are present in access specifiers in class ?
a) 4
b) 3
c) 2
d) 1

FAQs:

1. Write proper syntax and examples describing class and object in C++.

Ans a) Class : It is the building block that leads to OOP in C++. It is user-defined data type, which holds its own data members and member functions, which can be accessed and used by creating an instance of that class. A C++ class is like a blueprint for an object.

Syntax:

```
class ClassName
{ Access specifier;        // can be public, private, protected
  Data members;            // variables to be used.
  Member functions(){}     // methods to access data member
};                         // class name ends with semicolon.
```

Example:

```
Class student
{
  Public:
    int RnO;
    Float Percentage;
    void getdetails();
};
student s;
```

b) Objects: It is an instance of a class. When class is defined, no memory is allocated but when it is initiated, memory is allocated.

Syntax: Class Name Object Name;

Example:
```
Class student
{ Public:
    int Rno;
    string name;
};
int main()
{
  Student s;      // Class Name Object name.
  cin >> s.Rno >> name;
  Return 0;
}
```

2. What is difference between structure and class

Ans ① Class is a blueprint that defines the variables and the methods common to all object of a certain kind whereas a structure is a value type data type that can hold related data of various data type.

② Class can inherit from other classes or structures whereas structures can inherit.

③ Class can have a destructor whereas structure can't have destructor.

④ Class instantiaties an object using a new keyword whereas structure instantiates an object without using new keyword.

⑤ Class instances of is an object whereas instance of a structure is a structure variable.

⑥ "Class" defines class whereas "struct" defines structure.

⑦ If there are no access specifiers declared, then the members are private in class whereas public in structure.

MIT SCHOOL OF ENGINEERING

Rajbaug, Loni-Kalbhor, Pune

MIT-ADT UNIVERSITY
PUNE, INDIA

Name: Saura Toshniwal Class: PY-CSE Batch: A2

Subject: OOP Roll No.: 2213047 Exp. No.: 2

Name of the Experiment: Object and Class.

Performed on:

Submitted on:

Marks	Teacher's Signature with date
10	

MCQ

1. which statement of the following is true?
   a) Data members of a class accessed using dot operator.
   b) Data members of a class accessed using arrow
   c) Data members of a class using none of dot & arrow
   d) Data members of a class accessed using dot, arrow or direct call.

2. Which keyword is used to make data members have the value?
   a) double
   b) static
   c) Const
   d) abstract.

3. How many data members can a class contain?
   a) 32
   b) 255

Incomplete for Theory ☐ Diagrams ☐ Observation Table ☐
Calculations ☐ Graphs ☐ Results ☐ Conclusion ☐
Understanding Through Q/A ☐ Late Submission ☐ Neatness ☐

Teacher's Signature

c) 1024
d) As many as required.

4. The variables declared inside the class are known as data members and functions are known as.
a) data functions
b) Member functions
c) Member variables
d) Inline functions.

5. A —— member function can only be called by another function that is a member of its class.
a) Public
b) friend
c) private
d) static

FAQs:

1. A Bank account holder having data members name, acc number, date of birth, User Id, Password and Balance choose appropriate access specifier for the above data members and explain

Ans. Private access specifier will be appropriate for the given data members as these data members will contain users personal bank details so to keep it secure we will use private access

specifier and when these details will be required we will use member functions that are under public access specifier.

2) Explain the concept of class and object.

Ans. Object: They have states and behaviours. Eg: A dog has states-colour, name, breed as well as behaviour-wagging the tail, barking, eating. An Object is an instance of a class.

Class: A class can be defined as a template/blueprint that describes the behaviour/state that the object of it's type support.

Eg:
```
# include <iostream>
using namespace std;
Class Dog {
Public:
    String name, breed, colour;
    int age;
void barking () {
}
Void hungry () {
}
};
int main () {
Dog d;
cin >> d.name >> d.breed >> d.age >> d.colour;
d.barking();
d.hungry();
return 0;
}
```

MIT SCHOOL OF ENGINEERING

Rajbaug, Loni-Kalbhor, Pune

MIT-ADT UNIVERSITY PUNE, INDIA

Name: Sourav Toshniwal Class: FY-CSE Batch: A2

Subject: OOP Roll No.: 2213047 Exp. No.: 3

Name of the Experiment: Arithmetic operators and inheritance

Marks	Teacher's Signature with date
10	

Performed on:

Submitted on:

MCQs:

1) What is the precedence of Arithmetic operators?

✓a. %, *, /, +, -

b. %, +, /, *, -

c. +, -, %, *, /

d. %, +, -, *, /

2) What is the output of this program?

```
# include <iostream>
using namespace std;
int main (){
int a;
a = 5+3*5;
cout <<a ;
return 0;
}
```

a. 35 c. 25

✓b. 20 d. 30

Incomplete for Theory ☐ Diagrams ☐ Observation Table ☐

Calculations ☐ Graphs ☐ Results ☐ Conclusion ☐

Understanding Through Q/A ☐ Late Submission ☐ Neatness ☐

Teacher's Signature

3) Which programming language doesn't support multiple inheritance?

a. C++ & Java
b. C & C++
c. Java & small talk
d. Java

4. Which among the following is correct for a hierarchical inheritance?

a. Two base classes can be used to be derived into single class
b. Two or more classes can be derived into one class.
c. One base class can be derived into other two or more derived classes.
d. One base class can be derived into only 2 classes.

5. Which type of inheritance leads to diamond problem?

a. Single level
b. Multi-level
c. Multiple
d. Heirarchial

6. While inheritering a class, if no access mode is specified then which among the following is True? (is C++)

a. It gets inherited publicly by default.
b. It gets inherited protected by default.
c. It gets inherited privately by default
d. Not possible.

FAQ's

1) What are C++ access modifiers? Why use access modifiers in C++?

Ans a) The access modifiers of C++ are public, private & protected.

b) One of the main features of OOP languages such as C++ is data hiding.

c) Data hiding refers to restricting access to data members of a class. This is to prevent other functions and classes from tampering with the class data.

2) What are Base and derived class? What are the advantages of inheritance?

Ans a) The class whose members are inherited is called the base class, and the class that inherits those members is called the derived class.

b) The main advantages of inheritance are code reusability and readability.

MIT SCHOOL OF ENGINEERING

Rajbaug, Loni-Kalbhor, Pune

MIT-ADT UNIVERSITY PUNE, INDIA

Name: Sourav Toshniwal Class: FY-CSE Batch: A2

Subject: OOP Roll No.: 2213047 Exp. No.: 4

Name of the Experiment: Constructor.

Performed on:

Submitted on:

Marks	Teacher's Signature with date
10	

MCQs:

1) What is the role of a constructor in classes?

a. To modify the data whenever required.

b. To destroy an object.

c. To initialize the data members of an object when it is created

d. To call private functions from the outer world.

2) What is copy constructor?

a. A constructor that allows a user to move data from one object to another

b. A constructor to initialize an object withe the values of another object

c. A constructor to check the whether to objects are equal or not.

d. A constructor to kill other copies of a given object.

3) How many destructors are allowed in a class?

a. 1

b. 2

c. 3

d. any number.

Incomplete for Theory ☐ Diagrams ☐ Observation Table ☐

Calculations ☐ Graphs ☐ Results ☐ Conclusion ☐

Understanding Through Q/A ☐ Late Submission ☐ Neatness ☐

Teacher's Signature

4) State whether the following statements about the constructor are True or False.

i) Constructors should be declared in the private section

ii) Constructors are invoked automatically when the objects are created

a. True, True

b. True, False

c. False, True

d. False, False

5) Which of the following is true about constructors.

i) They cannot be virtual.

ii) They cannot be private.

iii) They are automatically called by new operator.

a. i, ii, iii

b. i & iii

c. ii & iii

d. i & ii

6) What will be the output

```
#include <iostream>
using namespace std;
class LFC
{
  int id;
  static int count;
  public:
  LFC ()
  {
    count ++;
```

```
    id=count;
    cout<<"constructor for id"<<id<<endl;
  }
~LFC ()
  { cout<<" destructor for id"<<id<<endl;
  }
};
int LFC ::count=0;
int main()
{ LFC a[3];
  return 0;
}
```

Ans: Constructor for id 1
constructor for id 2
Constructor for id 3
destructor for id 3
destructor for id 2
destructor for id 1

FAQ's :

1) What are the benefits of constructor & destructor ?

Ans a) Constructor: It is used to iniatize an object of the class & assign values to data members correspoding to the class.

b) Destructor: It is used to deallocate the memory of an object of a class. There can be multiple construction for the same class. In a class, there is always a single destructor.

2. Can I define my own constructor?

Ans. A constructor is a special member function that is public, has the same name as the class it is a part of it has no return type. It is used to initialize the members of a class. It can be of three types:

a. defaut constructor.
b. copy constructor.
c. parameterise constructor.

As a constructor always has the same name as the class, we cannot define it with another name.

Syntax:

```
Class ABC
{
  Public:
  ABC (parameter(s) if required)
  {
    // body
  }
};
```

MIT SCHOOL OF ENGINEERING
Rajbaug, Loni-Kalbhor, Pune

MIT-ADT UNIVERSITY PUNE, INDIA

Name: Sourav Toshniwal Class: FY-CSE Batch: A2
Subject: OOP Roll No.: 22B047 Exp. No.: 5
Name of the Experiment: Inheritance and Exception handling.
Performed on:
Submitted on:

Marks	Teacher's Signature with date
10	

MCQ's:

1) What is output?

```
#include <iostream>
using namespace std;
int main ()
{
  try
  { throw 20;
  }
  catch (int e)
  {
  cout<<"An exception occured"<< e << endl;
  }
return 0;
}.
```

a. 20 b. An exception occured

c. Error

d. An exception occured 20

Incomplete for Theory ☐ Diagrams ☐ Observation Table ☐
Calculations ☐ Graphs ☐ Results ☐ Conclusion ☐
Understanding Through Q/A ☐ Late Submission ☐ Neatness ☐

Teacher's Signature

2) Which block should be placed after the try block?
a. Catch
✓ b. throw
c. either catch or throw
d. none of the above

3) How do you define the user-defined exception?
a. Inheriting & overriding exception class functionality
b. Overriding class functionality.
✓ c. Inheriting class functionality.
d. None of the above.

4) Which keyword is used to handle the exception?
✓ a. Try
b. Throw
c. Catch
d. none of the above

FAQ's:

1) What is exception handling? What is the try catch blocks of exception handling?

Ans Exception handling is a mechanism that seperates code that detects and handles exceptional circumstances from the rest of your program. Note that an exceptional circumstance is not necessarily an error.

The try block contains set of statements where an exception can occurs. A try block is always followed by a catch block, which handles the exception that occurs in which handles the

exception that occurs in associated try block. A try block must be followed by catch blocks or finally block or both.

2) What is the difference between exception and error?

Ans Exception: An exception "indicates condition that a reasonable application might want to catch".

Error: An error "indicates serious problem that a reasonable application should not try to catch."

Error along with runtime exception & their subclasses are unchecked exception.

3) What will happen if an exception is not handled?

Ans The program terminates abruptly and the code past the line that caused the exception will not get executed.

MIT SCHOOL OF ENGINEERING

Rajbaug, Loni-Kalbhor, Pune

MIT-ADT UNIVERSITY PUNE, INDIA

Name: Sourav Toshniwal Class: FY-CSE Batch: A2

Subject: OOP Roll No.: 2213047 Exp. No.: 6

Name of the Experiment: Function Overloading (polymorphism).

Performed on:

Submitted on:

Marks	Teacher's Signature with date
10	

MCQ's:

1) Overloading functions are ___

a. Very long functions that can hardly run.

b. One function containing another one or more functions inside it

c. Two or more functions with the same name but different number of parameters or type

d. Very long functions.

2)
```
#include <iostream>
using namespace std;
int operate (int a, int b) {
  return (a*b);
  }
float operate (float a, float b) {
  return (a/b);
  }
int main() {
  int x=5, y=2;
```

Incomplete for Theory ☐ Diagrams ☐ Observation Table ☐

Calculations ☐ Graphs ☐ Results ☐ Conclusion ☐

Understanding Through Q/A ☐ Late Submission ☐ Neatness ☐

Teacher's Signature

```
float n=5.0, m=2.0;
cout << operate (x, y)<<"\t";
cout << operate (n, m);
return 0;
}
```

a. 10.0 5.0

b. 5.0 2.5

c. 10.0 5

d. 10 2.5

3) What will be the output of the following C++ code?

```
# include <iostream>
using namespace std;
int Add (int X, int Y, int Z) {
  return X+Y;
  }
double Add (double X, double Y, double Z) {
  return X+Y;
  }
int main (){
  cout << Add (5,6);
  cout << Add (5.5, 6.6);
  return 0;
  }
```

a. 11 12.1

b. 12.1 11

c. 11 12

d. Compile time error

4) Function overloading is also similar to which of the following?
a. Operator overloading.
b. constructor overloading
c. destructor overloding.
d. function overloading.

5) In which of the following we cannot overload the function?
a. return function
b. caller
c. called function
d. main function.

FAQ's:

1) How function overloading is different from function overriding?

Ans a. Function overloading is when multiple function with same name exist in a class. Function overriding is when function have same prototype in base as well as derived class.
b. function Overloading can occur without inheritance. Function Overriding occurs when one class is inherited from another class.
c. Overloaded functions must be different in either number of parameters or type of parameters should be different. In Overridden function parameters must be same.

2) What is function overloading? Explain with example. What is the need of function overloading in C++? What are the limitations of function overloading?

Ans: Function overloading is the process of creating multiple functions of the same name while giving them different parameters/return types. It is a feature of OOP that is also known as compile time polymorphism. The syntax of function overloading define as follows.

return - type 1 function-name (parameter 1);
return - type 2 function-name (parameter 2);

Eg:

```
class print
{
  void print (int);
  void print (float);
  void print (string);
};
```

We need function overloading in C++ as it helps us save memory resources and improves responsibility and readability of the code. It speeds up the execution of the program & helps make code maintainance easier.

Some limitations of function overloading are:
Function declarations that differ only by its return type can't be overloaded with function overloading process. Member function declarations with the same name types can't be overloaded if any one of them is declared as a static member function. Using function overloading can lead to ambiguity arising in some cases.

MIT SCHOOL OF ENGINEERING

Rajbaug, Loni-Kalbhor, Pune

MIT-ADT UNIVERSITY PUNE, INDIA

Name: Sourav Toshniwal Class: A Batch: A2

Subject: OOP Roll No.: 2213047 Exp. No.: 7

Name of the Experiment: Operator overloading.

Marks	Teacher's Signature with date
10	

Performed on:

Submitted on:

MCQs:

1) Which of the following operator cannot be overloaded?

a) . (Member access or dot operator)

b) ?: (Ternary or Conditional Operator)

c) :: (Scope Resolution Operator)

d) .* (Pointer - to - member Operator)

e) All of the above.

2) Which of the following operators are overloaded by default by the compiler in every user defined classes even if user has not written?

1) Comparison Operator (==)

2) Assignment operator (=)

a) Both 1 & 2

b) Only 1

c) Only 2

d) None of the two

Incomplete for Theory ☐ Diagrams ☐ Observation Table ☐

Calculations ☐ Graphs ☐ Results ☐ Conclusion ☐

Understanding Through Q/A ☐ Late Submission ☐ Neatness ☐

Teacher's Signature

3) Which of the following operators should be prefered to overload as a global function rather than a member method?
a) Postfix ++
b) Comparison Operator
c) Insertion Operation <<
d) Prefix ++

4) Which of the following operator functions cannot be global, i.e. must be a member function.
a) New
b) delete
c) Conversion Operator
d) All of the above.

5) Which is the correct statement about operator overloading in C++?
a) Only arithmetic operators can be overloaded
b) Associativity and precedence of operators does not change
c) Precedence of operators are changed after overloading.
d) Only non-arithmetic operators can be overloaded.

FAQ's :

1) What are the benefits of operator overloading?

Ans a) Operator overloading enables programmers to use notation closer to target domain.

b) Operator overloading provides synthetic support of build-in types in types to user defined types.

c) You can give new / complex meaning to simple operator to do complex calculations.

2) Can I define my own operators?

Ans Yes by using operator overloading we can give define to pre existing operators :

Eg :

```
Class Class_name
{
  public :
  return_type operator symbol (argument)
  {
    // new meaning to pre-existing operator
  }
};
```

MIT SCHOOL OF ENGINEERING

Rajbaug, Loni-Kalbhor, Pune

MIT-ADT UNIVERSITY PUNE, INDIA

Name: Sourav Toshniwal Class: P4-CSE Batch: A2

Subject: OOP Roll No.: 2213047 Exp. No.: 8

Name of the Experiment: Class template.

Performed on:

Submitted on:

Marks	Teacher's Signature with date
10	

MCQ's:

1) What is a template?

a) A template is a way of creating a generic class

b) A template is used to manipulate the class.

c) A template is used for creating the attributes

d) None of the above.

2) What is meant by the template parameter?

a) It can be used to pass a type as an argument

b) It can be used to evaluate a type

c) It can of no return type.

d) It can be used to delete a type.

3) Which keyword can be used in template?

a) Class

b) typename

c) both class & typename

d) function.

Incomplete for Theory ☐ Diagrams ☐ Observation Table ☐

Calculations ☐ Graphs ☐ Results ☐ Conclusion ☐

Understanding Through Q/A ☐ Late Submission ☐ Neatness ☐

Teacher's Signature

4) What will be the output of the following C++ code?

```
#include <iostream>
Using namespace std;
template < class T, int N>
Class mysequence
{
  T memblock [N];
  public :
  void setmember (int X, T value);
  T getmemble (int X);
};
template <class T , int N >
void mysequence <T, N> :: setmember (int X, T value)
{
  memblock [X] = value ;
}
template < Class T, int N>
T mysequence <T, N> :: getmember (int X)
{
  return memblock [X];
}
int main ()
{
mysequence <int , 5 > myints ;
mysequence < double , 5> myfloats ;
myints . setmember (0, 100);
myfloats . setmember (3, 3. 1416);
cout << myints . getmember (0) << '\n';
```

```
cout << myfloats.get member(3) << '\n';
return 0;
}
```

a) 100
b) 3.1416
c) 100 3.1416
d) 4.14

5) — allows programmers to define generic classes & function & thus provide support for generic programming.
a) Super class
b) templates
c) Containers
d) Polymorphism.

FAQ's.

1) What is template in C++. Types of Templates.

Ans Templates in C++ is defined as blueprint or formula for creating a generic class or a function. To simply put, you can create a single function or single class to work with different data types using templates.
C++ template is also known as generic functions or classes which is a very powerful feature in C++.
Types of templates in C++
- function template
- Class template.

2) Explain function template and class template with example.

Ans Function template in C++ is a single function template that works with multiple datatypes simultaneously. but a standard function works only with one set of data types.

SYNTAX:

```
template < class type > ret-type function name.
{
  // body of the function
}
```

Class template in C++ is like function templates. They are known as generic templates. They define a family of classes in C++.

SYNTAX:

```
templete < class Type >
Class class-name
{
  // class body;
}.
```

MIT SCHOOL OF ENGINEERING

Rajbaug, Loni-Kalbhor, Pune

MIT-ADT UNIVERSITY PUNE, INDIA

Name: Sourav Toshniwal Class: FY-CSE A Batch: A2

Subject: OOP Roll No.: 2213047 Exp. No.: 9

Name of the Experiment: Different OOP languages.

Performed on:

Submitted on:

Marks	Teacher's Signature with date
10	

MCQ's

1) Which of the following is use to take the inform user in JAVA?

a) Cout

b) scanner (System.In)

c) Java.util

d) Cin.

2) Which of the following is use to take the inform user in Python?

a) Cout

b) scanner (system.in)

c) Python input ()

d) python.in

3) Which of the following language was developed as the first purely OOP language?

a) Smalltalk

Incomplete for Theory ☐ Diagrams ☐ Observation Table ☐

Calculations ☐ Graphs ☐ Results ☐ Conclusion ☐

Understanding Through Q/A ☐ Late Submission ☐ Neatness ☐

Teacher's Signature

b) java
c) python
d) cpp

4) which among the following feature is not in the general OOPs ?
a) Modularity.
b) Efficient Code
c) Code reusability.
d) Duplicate or Reducdant Data.

5) If you are using any OOPs programming language, whole code of program must same be for different programming language.
a) True
b) False.

FAQ's

1) What is the main key difference between C++ and Java?

Ans C++ :- It is platform dependent hence it is not portable
- It is a compiled language.
- It supports both multiple and single inheritance.
- It is both procedural and an object-oriented programming language.

Java :- It can run in any OS hence it is portable
- It is both compiled and interpreted language.
- It supports only single inheritance. Multiple inheritance are achieved partially using interface.
- It is only an Object Oriented programming language.

2) What are similarities and differences between Java, C++ and python?

Ans C++ :- Compiled programming language.
- Support operator overloading.
- Provide with single and multiple inheritance.

Python :- Interpreted programming language.
- Support operator overloading.
- Provide both single & multiple inheritance.

Java :- Compiled programming language
- Does not support operator overloading
- Provide partial multiple inheritance using interface.