

MCQs for REFERENCE

Q. 1 What is the output of this program?

```
1. #include <iostream>
2. using namespace std;
3. template <class T, int N>
4. class mysequence
5. {
6.     T memblock [N];
7.     public:
8.     void setmember (int x, T value);
9.     T getmember (int x);
10. };
11. template <class T, int N>
12. void mysequence<T,N> :: setmember (int x, T value)
13. {
14.     memblock[x] = value;
15. }
16. template <class T, int N>
17. T mysequence<T,N> :: getmember (int x)
18. {
19.     return memblock[x];
20. }
21. int main ()
22. {
23.     mysequence <int, 5> myints;
24.     mysequence <double, 5> myfloats;
25.     myints.setmember (0, 100);
26.     myfloats.setmember (3, 3.1416);
27.     cout << myints.getmember(0) << '\n';
28.     cout << myfloats.getmember(3) << '\n';
29.     return 0;
30. }
```

A 100

B 3.1416

C 100

3.1416

D none of the mentioned

E

Ans. C

Q. 2 What is the output of this program?

```
1. #include <iostream>
2. using namespace std;
3. template <class T>
4. T max (T a, T b)
5. {
6.     return (a>b?a:b);
7. }
```

MCQs for REFERENCE

```
8. int main ()
9. {
10.     int i = 5, j = 6, k;
11.     long l = 10, m = 5, n;
12.     k = max(i, j);
13.     n = max(l, m);
14.     cout << k << endl;
15.     cout << n << endl;
16.     return 0;
17. }
```

A a) 6

B b) 6 10

C c) 5 10

D d) 6 5

Ans. B

Q. 3 What is the output of this program?

```
1. #include <iostream>
2. using namespace std;
3. template <class type>
4. class Test
5. {
6.     public:
7.     Test()
8.     {
9.     };
10.    ~Test()
11.    {
12.    };
13.    type Funct1(type Var1)
14.    {
15.        return Var1;
16.    }
17.    type Funct2(type Var2)
18.    {
19.        return Var2;
20.    }
21. };
22. int main()
23. {
24.     Test<int> Var1;
25.     Test<double> Var2;
26.     cout << Var1.Funct1(200);
27.     cout << Var2.Funct2(3.123);
28.     return 0;
29. }
```

MCQs for REFERENCE

- A 100
 - B 200
 - C 3.123
 - D 200 3.123
 - E
- Ans. D

Q. 4 What is the output of this program?

```
1. #include <iostream>
2. using namespace std;
3. template <typename T, int count>
4. void loopIt(T x)
5. {
6.     T val[count];
7.     for(int ii = 0; ii < count; ii++)
8.     {
9.         val[ii] = x++;
10.        cout << val[ii] << endl;
11.    }
12. };
13. int main()
14. {
15.     float xx = 2.1;
16.     loopIt<float, 3>(xx);
17. }
```

- A 2.1
 - B 3.1
 - C 4.1
 - D 2.1
3.1
4.1
- Ans. D

Q. 5 What is the output of this program?

```
1. #include <iostream>
3. int main()
4. {
5.     int x = -1;
6.     try
7.     {
8.         if (x < 0)
9.         {
10.            throw x;
11.        }
12.        else
13.        {
14.            cout<<x;
15.        }
16.    }
```

MCQs for REFERENCE

```
17. catch (int x )
18. {
19.     cout << "Exception occurred: Thrown value is " << x << endl;
20. }
21. return 0;
22. }
```

- A a) -1
B b) 0
C Exception occurred: Thrown value is -1
D error
E
Ans. C

Q. 6 What is the output of this program?

```
1. #include <iostream>
2. #include <typeinfo>
3. using namespace std;
4. class Polymorphic {virtual void Member(){} };
5. int main ()
6. {
7.     try
8.     {
9.         Polymorphic * pb = 0;
10.        typeid(*pb);
11.    }
12.    catch (exception& e)
13.    {
14.        cerr << "exception caught: " << e.what() << endl;
15.    }
16.    return 0;
17. }
```

- A exception caught: std::bad_typeid
B exception caught: std::bad_alloc
C exception caught: std::bad_cast
D none of the mentioned
E
Ans. A

Q. 7 What is the output of this program?

```
1. #include <iostream>
2. #include <exception>
3. using namespace std;
4. void myunexpected ()
5. {
6.     cout << "unexpected handler called\n";
7.     throw;
8. }
9. void myfunction () throw (int,bad_exception)
```

MCQs for REFERENCE

```
10. {
11.     throw 'x';
12. }
13. int main (void)
14. {
15.     set_unexpected (myunexpected);
16.     try
17.     {
18.         myfunction();
19.     }
20.     catch (int)
21.     {
22.         cout << "caught int\n";
23.     }
24.     catch (bad_exception be)
25.     {
26.         cout << "caught bad_exception\n";
27.     }
28.     catch (...)
29.     {
30.         cout << "caught other exception \n";
31.     }
32.     return 0;
33. }
```

- A unexpected handler called
- B caught bad_exception
- C caught other exception
- D both a & b

Ans. D

Q. 8 What is the output of this program?

```
1. #include <iostream>
2. using namespace std;
3. int main()
4. {
5.     int x = -1;
6.     char *ptr;
7.     ptr = new char[256];
8.     try
9.     {
10.        if (x < 0)
11.        {
12.            throw x;
13.        }
14.        if (ptr == NULL)
15.        {
16.            throw " ptr is NULL ";
17.        }
18.    }
```

MCQs for REFERENCE

```
19. catch (...)  
20. {  
21.     cout << "Exception occurred: exiting " << endl;  
22. }  
23. return 0;  
24. }
```

A

B

C

D

E

Ans.

-1
ptr is NULL
Exception occurred: exiting
none of the mentioned

C

Q. 9

What is the output of this program?

```
1. #include <iostream>  
2. #include <exception>  
3. using namespace std;  
4. void myunexpected ()  
5. {  
6.     cout << "unexpected called\n";  
7.     throw 0;  
8. }  
9. void myfunction () throw (int)  
10. {  
11.     throw 'x';  
12. }  
13. int main ()  
14. {  
15.     set_unexpected (myunexpected);  
16.     try  
17.     {  
18.         myfunction();  
19.     }  
20.     catch (int)  
21.     {  
22.         cout << "caught int\n";  
23.     }  
24.     catch (...)  
25.     {  
26.         cout << "caught other exception\n";  
27.     }  
28.     return 0;  
29. }
```

A

B

C

D

E

Ans.

caught other exception
caught int
unexpected called
both b & d

D

MCQs for REFERENCE

Q. 10

What is the output of this program?

```
#include <iostream>
#include <fstream>
using namespace std;
int main ()
{
    int length;
    char * buffer;
    ifstream is;
    is.open ("sample.txt", ios :: binary );
    is.seekg (0, ios :: end);
    length = is.tellg();
    is.seekg (0, ios :: beg);
    buffer = new char [length];
    is.read (buffer, length);
    is.close();
    cout.write (buffer, length);
    delete[] buffer;
    return 0;
}
```

A

This is sample

B

sample

C

Error

D

Runtime error

E

Ans.

D

Q. 11

What is the output of this program?

```
#include<iostream>
#include <fstream>
using namespace std;
int main ()
{
    ofstream outfile ("test.txt");
    for (int n = 0; n < 100; n++)
    {
        outfile << n;
        outfile.flush();
    }
    cout << "Done";
    outfile.close();
    return 0;
}
```

A

Done

B

Error

C

Runtime error

D

None of the mentioned

E

MCQs for REFERENCE

Ans.

A

Q. 12

What is the output of this program?

```
#include <iostream>
using namespace std;
int main ()
{
    int a = 100;
    double b = 3.14;
    cout << a;
    cout << endl;
    cout << b << endl << a * b;
    endl (cout);
    return 0;
}
```

A

100

B

3.14

C

314

D

All of the mentioned

E

Ans.

D

Q. 13

By seeing which operator thus this program stops getting the input?

```
#include <iostream>
#include <fstream>
using namespace std;
int main ()
{
    char ch;
    streambuf * p;
    ofstream os ("test.txt");
    pbuf = os.rdbuf();
    do {
        ch = cin.get();
        p -> sputc(ch);
    } while (ch != '.');
    os.close();
    return 0;
}
```

A

dot operator

B

insertion operator

C

\$ symbol

D

None of the mentioned

E

Ans.

A

Q. 14

What is the output of this program?

MCQs for REFERENCE

```
#include <iostream>
#include <exception>
using namespace std;
class base { virtual void dummy() {} };
class derived: public base { int a; };
int main ()
{
    try
    {
        base * pba = new derived;
        base * pbb = new base;
        derived * pd;
        pd = dynamic_cast<derived*>(pba);
        if (pd == 0)
            cout << "Null pointer on first type-cast" << endl;
        pd = dynamic_cast<derived*>(pbb);
        if (pd == 0)
            cout << "Null pointer on second type-cast" << endl;
    }
    catch (exception& e)
    {
        cout << "Exception: " << e.what();
    }
    return 0;
}
```

- A Null pointer on first type-cast
- B Null pointer on second type-cast
- C Exception
- D None of the mentioned

Ans.

B

Q. 15

What is the output of this program?

```
#include <iostream>
#include <typeinfo>
using namespace std;
int main ()
{
    int * a;
    int b;
    a = 0; b = 0;
    if (typeid(a) != typeid(b))
    {
        cout << typeid(a).name();
        cout << typeid(b).name();
    }
    return 0;
}
```

- A
- B

int *
int

MCQs for REFERENCE

C Both a & b
D float
E
Ans. C

Q. 16 What is the output of this program?

```
#include <iostream>
#include <typeinfo>
#include <exception>
using namespace std;
class base
{
    virtual void f(){}
};
class derived : public base {};
int main ()
{
    try
    {
        base* a = new base;
        base* b = new derived;
        cout << typeid(*a).name() << '\t';
        cout << typeid(*b).name();
    }
    catch (exception& e)
    {
        cout << "Exception: " << e.what() << endl;
    }
    return 0;
}
```

A base*
B derived*
C base and derived
D None of the mentioned
E
Ans. C

Q. 17 What is the output of this program?

```
#include <typeinfo>
#include <iostream>
using namespace std;
class A
{
    public:
    virtual ~A();
};
int main()
{
    A* a = NULL;
```

MCQs for REFERENCE

```
try
{
    cout << typeid(*a).name() << endl;
}
catch (bad_typeid)
{
    cout << "Object is NULL" << endl;
}
}
```

- A
 - B
 - C
 - D
 - E
- Ans.

int
float
double
object is NULL
D

Q. 18

What is the output of this program?

```
#include <iostream>
using namespace std;
struct A
{
    virtual void f()
    {
        cout << "Class A" << endl;
    }
};
struct B : A
{
    virtual void f()
    {
        cout << "Class B" << endl;
    }
};
struct C : A
{
    virtual void f()
    {
        cout << "Class C" << endl;
    }
};
void f(A* arg)
{
    B* bp = dynamic_cast<B*>(arg);
    C* cp = dynamic_cast<C*>(arg);
    if (bp)
        bp -> f();
    else if (cp)
        cp -> f();
    else
        arg -> f();
}
```

MCQs for REFERENCE

```
};  
int main()  
{  
    A aobj;  
    C cobj;  
    A* ap = &cobj;  
    A* ap2 = &aobj;  
    f(ap);  
    f(ap2);  
}
```

- A Class C
 - B Class A
 - C Both a & b
 - D None of the mentioned
 - E
- Ans. C

- Q. 19 What is meant by template parameter?
- A It can be used to pass a type as argument
 - B It can be used to evaluate a type.
 - C It can of no return type
 - D None of the mentioned
 - E
- Ans. A

- Q. 20 What is the validity of template parameters?
- A inside that block only
 - B inside the class
 - C whole program
 - D any of the mentioned
 - E
- Ans. A

- Q. 21 Why we use :: template-template parameter?
- A Binding
 - B Rebinding
 - C both a & b
 - D none of these
 - E
- Ans. C

- Q. 22 Which parameter is legal for non-type template?
- A pointer to member
 - B object
 - C class
 - D none of the mentioned
 - E
- Ans. A

- Q. 23 Which of the things does not require instantiation?

MCQs for REFERENCE

- A functions
B non virtual member function
C member class
D all of the mentioned
Ans. D
- Q. 24 Which is used to handle the exceptions in c++?
A catch handler
B handler
C exceptional handler
D none of the mentioned
Ans. C
- Q. 25 Which type of program is recommended to include in try block?
A static memory allocation
B dynamic memory allocation
C const reference
D Pointer
Ans. B
- Q. 26 Which statement is used to catch all types of exceptions?
A catch()
B catch (Test t)
C catch(...)
D none of the mentioned
Ans. C
- Q. 27 How to handle error in the destructor?
A throwing
B terminate
C both a & b
D none of the mentioned
Ans. **B**
- Q. 28 What kind of exceptions are available in c++?
A handled
B unhandled
C static
D dynamic
Ans. B
- Q. 29 An exception is caused by

MCQs for REFERENCE

- A A hardware problem
B A problem in the operating system
C A run-time error
D A syntax error
Ans. C
- Q. 30 An exception thrown from outside try block will _____.
A call function return
B be ignored
C hang the machine
D call function terminate
Ans. D
- Q. 31 To perform stream I/O with disk files in C++, you should
A open and close files as in procedural languages.
B use classes derived from ios.
C use C language library functions to read and write data.
D include the IOSTREAM.H header file.
Ans. B
- Q. 32 If we create a file by 'ifstream', then the default mode of the file is
A ios :: out
B ios :: in
C ios :: app
D None
Ans. **B**
- Q. 33 If we create a file by 'ofstream', then the default mode of the file is
A ios :: out
B ios :: in
C ios :: app
D None
Ans. A
- Q. 34 If we create a file by 'fstream', then the default mode of the file is
A ios :: out
B ios :: in
C ios :: app
D None
Ans. D
- Q. 35 seekg() is a member function of
A istream class
B ostream class

MCQs for REFERENCE

C file class
D None of the above
Ans. A

Q. 36 tellg() is a member function of

A istream class
B ostream class
C file class
D None of the above
Ans. A

Q. 37 How many bytes will be written to the file by the following code:

```
void main()
{
ofstream test("test2");
int A[20] = { 1, 2, 3 };
test.write((char*)A,10);
test.write((char*)A,sizeof(A));
test.close();
}
```

A 10
B 20
C 40
D 50
Ans. D

Q. 38 Which keyword can be used in template?

A class
B typename
C both a & b
D function
Ans. C

Q. 38 Object _____ and operator _____ are used for reading.

A cin >>
B cout >>
C cin <<
D cout <<
Ans. A

Q. 39 .Of the following, _____ is not a manipulator.

A dec
B scientific
C endl
D flush
Ans. **B**

Q. 40 Which operators are part of RTTI?

A dynamic_cast()

MCQs for REFERENCE

B typeid
C Both a & b
D None of the mentioned
Ans. C

Q. 41 What is meant by type_info?
A Used to hold the type information returned by the typeid operator
B Used to hold the type information returned by the dynamic_cast
C Used to hold the type information returned by the static cast
D None of the mentioned
Ans. A

Q. 42 At which time does the static_cast can be applied?
A Compile-time construct
B Runtime construct
C Both a & b
D None of the mentioned
Ans. A

A Encapsulation
B Polymorphic
C Derived
D None of the mentioned
Ans. **B**

Q. 44 What is the Run-Time Type Information?
A Information about an object's datatype at runtime
B Information about the variables
C Information about the given block
D None of the mentioned
Ans. A

Q. 45 Which member function is used to determine whether the stream object is currently associated with a file?
A is_open
B buf
C string
D None of the mentioned
E
Ans. A

Q. 46 How many bytes will be written to the file by the following code:

```
void main()
{
ofstream test("test");
char A[20] = "India";
test.write(A,20);
test.close();
}
```

A 5

MCQs for REFERENCE

B 6
C 19
D 20
E
Ans. D

Q. 47 The statement `f1.write((char*)&obj1, sizeof(obj1));`
A writes the member function of obj1 to f1.
B Writes the data in obj1 to f1.
C Writes the member function and the data of obj1 to f1.
D Writes the address of obj1 to f1.
Ans. B

Q. 48 `seekg()` returns:
A Current value of get pointer
B Current value of put pointer
C Number of records in the file
D None of the above
Ans. D

Q. 49 If we create a file by 'ofstream', then we can
A write data into the file
B Read data from the file
C Neither read from nor write into the file
D Read from as well as write into the file
Ans. A

Q. 50 To perform stream I/O with disk files in C++, you should
A open and close files as in procedural languages.
B use classes derived from ios.
C use C language library functions to read and write data.
D include the IOSTREAM.H header file.
Ans. B