# MULTIPLE CHOICE QUESTIONS

**Class : I B.Sc Computer Science**

# Subject : Object Oriented Programming with C++ Subject Code : U21CSC203

**UNIT – I**

1. Who invented C++?
   1. Dennis Ritchie b) Ken Thompson c) Brian Kernighan d) Bjarne Stroustrup Answer: d
2. What is C++?
3. C++ is an object oriented programming language
4. C++ is a procedural programming language
5. C++ supports both procedural and object oriented programming language
6. C++ is a functional programming language Answer: c
7. Which of the following type is provided by C++ but not C?
   1. double b) float c) int d) bool Answer: d
8. What is Inheritance in C++?
   1. Deriving new classes from existing classesb) Overloading of classes

c) Classes with same names d) Wrapping of data into a single class Answer: a

1. What is meant by a polymorphism in C++?
   1. class having only single form b) class having four forms

c) class having many forms d) class having two forms Answer: c

1. What is abstract class in C++?
   1. Any Class in C++ is an abstract class
   2. Class from which any class is derived
   3. Class specifically used as a base class with atleast one virtual functions
   4. Class specifically used as a base class with atleast one pure virtual functions Answer: d
2. Which concept allows you to reuse the written code in C++?
   1. Inheritance b) Polymorphism c) Abstraction d) Encapsulation Answer: a
3. Data is hidden and cannot be accessed by ?
   1. internal function b) external function c) inline function d) main function Answer: b
4. From which function the execution of a C++ program starts?
   1. start() function b) main() function c) new() function d) end() function Answer: b
5. A language which has the capability to generate new data types are called
   1. Extensible b) Overloaded c) Encapsulated d) Reprehensible Answer: a
6. Wrapping data and its related functionality into a single entity is known as
   1. Abstraction b) Encapsulation c) Polymorphism d) Modularity Answer: b
7. How structures and classes in C++ differ?
   1. In Structures, members are public by default whereas, in Classes, they are private by default
   2. In Structures, members are private by default whereas, in Classes, they are public by default
   3. Structures by default hide every member whereas classes do not
   4. Structures cannot have private members whereas classes can have Answer: a
8. What does polymorphism in OOPs mean?
   1. Concept of allowing overiding of functions
   2. Concept of hiding data
   3. Concept of keeping things in differnt modules/files
   4. Concept of wrapping things into a single unit Answer: a
9. Which concept allows you to reuse the written code?
   1. Encapsulation b) Abstraction c) Inheritance d) Polymorphism Answer: c
10. Which of the following is the correct syntax of including a user defined header files in C++?
    1. #include [userdefined] b) #include “userdefined”

c) #include <userdefined.h> d) #include <userdefined> Answer: b

1. Which of the following is used for comments in C++?
   1. /\* comment \*/ b) // comment \*/
2. // comment d) both // comment or /\* comment \*/ Answer: d
3. Which of the following user-defined header file extension used in c++?
   1. hg b) cpp c) h d) hf Answer: c
4. which of the following approach is used by c++?
   1. Left-right b) Right-left c) Bottom-up d) Top-down Answer: c
5. What is virtual inheritance in C++?
6. C++ technique to enhance multiple inheritance
7. C++ technique to ensure that a private member of the base class can be accessed somehow
8. C++ technique to avoid multiple inheritances of classes
9. C++ technique to avoid multiple copies of the base class into children/derived class Answer: d
10. What happens if the following C++ statement is compiled and executed? int\*ptr=NULL;

deleteptr;

1. The program is not semantically correct
2. The program is compiled and executed successfully
3. The program gives a compile-time error
4. The program compiled successfully but throws an error during run-time Answer: b
5. What will be the output of the following C++ code? #include <iostream>

#include <string> usingnamespacestd;

int main(intargc, charconst\*argv[])

{

char s1[6]="Hello"; char s2[6]="World";

char s3[12]= s1 +" "+ s2; cout<<s3;

return0;

}

1. Hello
2. World
3. Error
4. Hello World

Answer: c

1. What happens if the following program is executed in C and C++? #include <stdio.h>

int main(void)

{

intnew=5; printf("%d", new);

}

1. Error in C and successful execution in C++
2. Error in both C and C++
3. Error in C++ and successful execution in C
4. A successful run in both C and C++

Answer: c

1. What happens if the following program is executed in C and C++? #include <stdio.h>

voidfunc(void)

{

printf("Hello");

}

void main()

{

func();

func(2);

}

1. Outputs Hello twice in both C and C++
2. Error in C and successful execution in C++
3. Error in C++ and successful execution in C
4. **Error in both C and C++**

Answer: d

1. Which of the following explains Polymorphism? a)int func(int, int);

float func1(float, float); b)int func(int);

int func(int); c)int func(float);

float func(int, int, char);

1. int func(); int new\_func(); Answer: c
2. Which of the following shows multiple inheritances?
   1. A->B->C b) A->B; A->C c) A,B->C d) B->A Answer: c
3. How access specifiers in Class helps in Abstraction?
   1. They does not helps in any way
   2. They allows us to show only required things to outer world
   3. They help in keeping things together
   4. Abstraction concept is not used in classes Answer: b
4. C++ is
   1. procedural programming language
   2. object oriented programming language
   3. functional programming language
   4. both procedural and object oriented programming language Answer: d
5. What does modularity mean?
   1. Hiding part of program b) Subdividing program into small independent parts

c) Overriding parts of program d) Wrapping things into single unit Answer: b

1. Which concept means the addition of new components to a program as it runs?
   1. Data hiding b) Dynamic binding c) Dynamic loading d) Dynamic typing

Answer: c

1. Which of the following approach is used by C++?
   1. Top-down b) Bottom-up c) Left-right d) Right-left Answer: b
2. Which of the following supports the concept that reusability is a desirable feature of a language?
   1. It reduces the testing time b) It reduces maintenance cost

c) It decreases the compilation time d) It reduced both testing and maintenance time Answer: d

1. Which of the following is a static polymorphism mechanism?
   1. Function overloading b) Operator overloading c) Templates d) All of the mentioned

Answer: d

1. Which of the following is true?
2. All operators in C++ can be overloaded.
3. The basic meaning of an operator can be changed.
   1. I only b) II only c) Both I and II d) Neither I nor II Answer: d
4. Which of the following statement is true?
5. In Procedural programming languages, all function calls are resolved at compile-time
6. In Object Oriented programming languages, all function calls are resolved at compile-time
   1. I only b) II only c) Both I and II d) Neither I nor II Answer: a
7. Which feature of the OOPS gives the concept of reusability?
   1. Abstraction b)Encapsulation c)Inheritance d)None of the above. Answer: c
8. How structures and classes in C++ differ?
   1. Structures by default hide every member whereas classes do not
   2. In Structures, members are public by default whereas, in Classes, they are private by default
   3. Structures cannot have private members whereas classes can have
   4. In Structures, members are private by default whereas, in Classes, they are public by default

Answer: b

1. The OOPs concept in C++, exposing only necessary information to users or clients is known as a)Data hiding b) Encapsulation c) Hiding complexity d) Abstraction

Answer: d

A class is made abstract by declaring at least one of its functions as?

* 1. abstract classes b) pure virtual function c) abstract functions d) Interface Answer: b

1. Which is private member functions access scope?
   1. Member functions which can used outside the class
   2. Member functions which are accessible in derived class
   3. **Member functions which can only be used within the class**
   4. Member functions which can't be accessed inside the class Answer: c
2. Can main() function be made private?
   1. Yes, always
   2. Yes, if program doesn't contain any classes
   3. No, because main function is user defined
   4. No, never Answer: d
3. At what point of time a variable comes into existence in memory is determined by its
   1. Data type b) Storage class c) Scope d) All of the above Answer: b
4. Which of the following concepts is used to implement late binding?
   1. Static function b) Virtual function c) Const function d) Operator function Answer: b
5. For Cat and Animal class, correct way of inheritance is
   1. Class Cat: public Animal b) Class Animal: public Cat

c) both are correct way d) None is correct way Answer: a

1. Which is not a feature of OOP in general definitions?
   1. Efficient Code b) Code reusability c) Modularity d) Duplicate/Redundant data Answer: d
2. Which was the first purely object oriented programming language developed?
   1. Kotlin b) SmallTalk c) Java d) C++ Answer: b
3. Which feature of OOP indicates code reusability?
   1. Abstraction b) Polymorphism c) Encapsulation **d) Inheritance**

Answer: d

1. Which header file is required in C++ to use OOP?
   1. OOP can be used without using any header file b) stdlib.h

c) iostream.h d) stdio.h Answer: a

1. Which among the following doesn’t come under OOP concept?
   1. Data hiding b) Message passing

c) Platform independent d) Data binding Answer: c

1. Which is the correct syntax of inheritance?
   1. class base\_classname :access derived\_classname{ /\*define class body\*/ };
   2. class derived\_classname : access base\_classname{ /\*define class body\*/ };
   3. class derived\_classname : base\_classname{ /\*define class body\*/ };
   4. class base\_classname : derived\_classname{ /\*define class body\*/ }; Answer: b
2. The feature by which one object can interact with another object is
   1. Message reading b) Message Passing c) Data transfer d) Data Binding Answer: b
3. Which among the following, for a pure OOP language, is true?
   1. The language should follow at least 1 feature of OOP
   2. The language must follow only 3 features of OOP
   3. The language must follow all the rules of OOP
   4. The language should follow 3 or more features of OOP Answer: c
4. What is encapsulation in OOP?
   1. It is a way of combining various data members and member functions that operate on those data members into a single unit
   2. It is a way of combining various data members and member functions into a single unit which can operate on any data
   3. It is a way of combining various data members into a single unit
   4. It is a way of combining various member functions into a single unit View Answer

Answer: a

1. Which of the following is not true about polymorphism?
   1. Helps in redefining the same functionality
   2. Increases overhead of function definition always
   3. It is feature of OOP
   4. Ease in readability of program Answer: b
2. What is an abstraction in object-oriented programming?
   1. Hiding the implementation and showing only the features b) Hiding the important data

c) Hiding the implementation d) Showing the important data Answer: a

1. Which among the following can show polymorphism?
   1. Overloading && b) Overloading << c) Overloading || d) Overloading += Answer: b
2. In which access should a constructor be defined, so that object of the class can be created in any function?
   1. Any access specifier will work b) Private c) Public d) Protected Answer: c
3. Which access specifier is usually used for data members of a class?
   1. Protected b) Private c) Public d) Default Answer: b
4. How to access data members of a class?
   1. Dot, arrow or direct call b) Dot operator

c) Arrow operator d) Dot or arrow as required Answer: d

1. Which feature of OOP reduces the use of nested classes?
   1. Inheritance b) Binding c) Abstraction d) Encapsulation Answer: a
2. Which of the following is not a property of an object?
   1. Properties b) Names c) Identity d) Attributes Answer: b
3. Which among the following best describes the Inheritance?
   1. **Using the data and functions into derived segment**
   2. Using already defined functions in a programming language
   3. Using the code already written once
   4. Copying the code already written Answer: a

# UNIT – II

1. Which of the following is a correct identifier in C++?
   1. VAR\_1234 b) $var\_name c) 7VARNAME d) 7var\_name Answer: a
2. What is the difference between delete and delete[] in C++?
   1. delete is syntactically correct but delete[] is wrong and hence will give an error if used in any case
   2. delete is used to delete normal objects whereas delete[] is used to pointer objects
   3. delete is a keyword whereas delete[] is an identifier
   4. delete is used to delete single object whereas delete[] is used to multiple(array/pointer of) objects

Answer: d

1. Which of the following is correct about this pointer in C++?
   1. this pointer is passed as a hidden argument in all static variables of a class
   2. this pointer is passed as a hidden argument in all the functions of a class
   3. this pointer is passed as a hidden argument in all non-static functions of a class
   4. this pointer is passed as a hidden argument in all static functions of a class Answer: c
2. Which of the following C++ code will give error on compilation? code 1

#include <iostream> usingnamespacestd;

int main(intargc, charconst\*argv[])

{

cout<<"Hello World"; return0;

}

code 2

#include <iostream>

int main(intargc, charconst\*argv[])

{

std::cout<<"Hello World"; return0;

}

* 1. Code 1 only b) Neither code 1 nor code 2 c) Both code 1 and code 2 d) Code 2 only Answer: b

1. What is the value of p in the following C++ code snippet? #include <iostream>

usingnamespacestd; int main()

{

int p;

bool a =true; bool b =false; int x =10;

int y =5;

p =((x | y)+(a + b)); cout<< p;

return0;

}

a) 12 b) 0 c) 2 d) 16

Answer: d

1. What will be the output of the following C++ function? int main()

{

registerint i =1; int\*ptr=&i; cout<<\*ptr; return0;

}

* 1. Runtime error may be possible b) Compiler error may be possible

c) 1 d) 0

Answer: b

1. Which of the following correctly declares an array in C++?
   1. array{10}; b) array array[10]; c) int array; d) int array[10]; Answer: d
2. What will be the output of the following C++ code? #include<iostream>

usingnamespacestd; int main ()

{

intcin; cin>>cin;

cout<<"cin: "<<cin; return0;

}

* 1. Segmentation fault b) Nothing is printed c) Error d) cin: garbage value Answer: d

1. What is the use of the indentation in c++?
   1. distinguishes between comments and inner data
   2. distinguishes between comments and outer data
   3. distinguishes between comments and code
   4. distinguishes between comments and outer data Answer: c
2. Which is more effective while calling the C++ functions?
   1. call by object b) call by pointer c) call by value d) call by reference

Answer: d

1. What will be the output of the following C++ program? #include <iostream>

#include <string> #include <cstring> usingnamespacestd;

int main(intargc, charconst\*argv[])

{

constchar\*a ="Hello\0World"; cout<<a;

return0;

}

* 1. Hello b) World c) Error d) Hello World

Answer: a

1. Which of the following is used to terminate the function declaration in C++? a) ; b) ] c) ) d) :

Answer: a

1. What will be the output of the following C++ code? #include <iostream>

usingnamespacestd; int main()

{

char c =74; cout<< c; return0;

}

* 1. I b) J c) A d) N Answer: b

1. What will be the output of the following C++ program? #include <iomanip>

#include <iostream> usingnamespacestd; int main()

{

cout<<setprecision(17); double d =0.1;

cout<< d <<endl; return0;

}

a) compile time error b) 0.100001 c) 0.11 d) 0.10000000000000001 Answer: d

1. Which keyword is used to define the macros in c++?
   1. #macro b) #define c) macro d) define Answer: b
2. What is the correct syntax of accessing a static member of a class in C++? class A

{

public:

staticint value;

}

* 1. A->value b) A^value c) A.value d) A::value Answer: d

1. The C++ code which causes abnormal termination/behaviour of a program should be written under block.
   1. catch b) throw c) try d) finally Answer: c
2. What will be the output of the following C++ code? #include <iostream>

usingnamespacestd; int main()

{

int a =5; float b;

cout<<sizeof(++a + b); cout<< a;

return0;

}

a) 25 b) 45 c) 46 d) 26

Answer: b

1. Which of the following symbol is used to declare the preprocessor directives in C++?

a) $ b) ^ c) # d) \* Answer: c

1. What will be the output of the following C++ program? #include<iostream>

usingnamespacestd; int main()

{

int a =5;

auto check =[=]()

{

a =10;

};

check();

cout<<"Value of a: "<<a<<endl; return0;

}

* 1. Segmentation fault b) Value of a: 5 c) Value of a: 10 d) Error

Answer: d

1. What will be the output of the following C++ code? #include <iostream>

usingnamespacestd; void square (int\*x, int\*y)

{

\*x =(\*x)\*--(\*y);

}

int main ()

{

int number =30; square(&number, &number); cout<< number;

return0;

}

* 1. 30 b) Error c) Segmentation fault d) 870 Answer: d

1. What will be the output of the following C++ program? #include <iostream>

#include <string> usingnamespacestd; int main ()

{

std::stringstr("Welcome."); str.back()='!';

std::cout<<str<<endl; return0;

}

* 1. Welcome! b) Welcome!. c) Welcome. d) Welcome.! Answer: a

1. Pick the incorrect statement about inline functions in C++?
   1. Saves overhead of a return call from a function
   2. **They are generally very large and complicated function**
   3. These functions are inserted/substituted at the point of call
   4. They reduce function call overheads Answer: b
2. What will be the output of the following C++ program? #include <iostream>

usingnamespacestd; int main()

{

int n =5; void\*p =&n;

int\*pi =static\_cast<int\*>(p); cout<<\*pi <<endl;

return0;

}

* 1. 5 b) 6 c) compile time error d) runtime error Answer: a

1. What will be the output of the following C++ code snippet? #include <iostream>

usingnamespacestd; 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;

float n =5.0, m =2.0; cout<< operate(x, y)<<"\t"; cout<< operate (n, m); return0;

}

a) 10.0 5 b) 10 2.5 c) 10.0 5.0 d) 5.0 2.5

Answer: b

1. What will be the output of the following C++ code? #include <iostream>

usingnamespacestd; int main ()

{

int a, b, c; a =2;

b =7;

c =(a > b)? a : b; cout<< c; return0;

}

a) 12 b) 14 c) 6 d) 7

Answer: d

1. What will be the output of the following C++ code snippet? #include <stdio.h>

#include<iostream> usingnamespacestd; int main ()

{

int array[]={0, 2, 4, 6, 7, 5, 3};

int n, result =0; for(n =0; n <8; n++)

{

result += array[n];

}

cout<< result; return0;

}

a) 21 b) 27 c) 26 d) 25

Answer: b

1. What will be the output of the following C++ program? #include <iostream>

#include <string> usingnamespacestd; int main ()

{

string str("Computerscience"); for(size\_ti=0;i<str.length();)

{

cout<<str.at(i-1);

}

return0;

}

* 1. runtime error b) Sanfo c) S d) Computerscience Answer: a

1. Which operator is having the right to left associativity in the following?
   1. Array subscripting b) Function call

c) Addition and subtraction d) Type cast Answer: d

1. Which operator is having the highest precedence?
   1. postfix b) unary c) shift d) equality Answer: a
2. What is this operator called ?:?
   1. conditional b) relational c) casting operator d) unrelational Answer: a
3. What will be the output of the following C++ code? #include <iostream>

using namespace std; int main()

{

int a;

a = 5 + 3 \* 5;

cout << a; return 0;

}

a) 35 b) 20 c) 25 d) 30

Answer: b

1. What will be the output of the following C++ code? #include <iostream>

using namespace std; int main()

{

int a = 5, b = 6, c, d; c = a, b;

d = (a, b);

cout << c << ' ' << d; return 0;

}

a) 5 b) 6 c) 8 d) 6 Answer: a

1. What will be the output of the following C++ code? #include <iostream>

using namespace std; int main()

{

int i, j; j = 10;

i = (j++, j + 100, 999 + j);

cout << i; return 0;

}

a) 1000 b) 11 c) 1010 d) 1001

Answer: c

1. What will be the output of the following C++ code? #include <iostream>

using namespace std; int main ()

{

int x, y; x = 5;

y = ++x \* ++x; cout << x << y;

x = 5;

y = x++ \* ++x; cout << x << y; return 0;

}

a) 749735 b) 736749 c) 367497 d) 367597

Answer: a

1. What will be the output of the following C++ code? #include <iostream>

using namespace std; int main()

{

int a = 5, b = 6, c; c = (a > b) ? a : b; cout << c;

return 0;

}

a) 6 b) 5 c) 4 d) 7 Answer: a

1. What will be the output of the following C++ code? #include <iostream>

using namespace std; main()

{

double a = 21.09399; float b = 10.20;

int c ,d;

c = (int) a;

d = (int) b;

cout << c <<' '<< d; return 0;

}

a) 20 b) 10 c) 21 d) 10

Answer: c

1. Where does the execution of the program starts?
   1. user-defined function **b) main function**

c) void function d) else function Answer: b

1. What are mandatory parts in the function declaration?
   1. return type, function name b) return type, function name, parameters

c) parameters, function name d) parameters, variables Answer: a

1. Which of the following is used to terminate the function declaration? a) : b) ) c) ; d) ]

Answer: c

1. Which is more effective while calling the functions?
   1. call by value b) call by reference
2. c) call by pointer d) call by object Answer: b
3. What will be the output of the following C++ code?

#include <iostream> using namespace std; void mani()

void mani()

{

cout<<"hai";

}

int main()

{

mani(); return 0;

}

* 1. hai b) haihai c) compile time error d) runtime error Answer: c

1. What will be the output of the following C++ code? #include <iostream>

using namespace std; void fun(int x, int y)

{

x = 20;

y = 10;

}

int main()

{

int x = 10; fun(x, x); cout << x; return 0;

}

* 1. 10 b) 20 c) compile time error d) 30 Answer: a

1. What is the scope of the variable declared in the user defined function?
   1. whole program **b) only inside the {} block**

c) the main function d) header section Answer: b

1. How many minimum number of functions should be present in a C++ program for its execution? a) 0 b) 1 c) 2 d) 3

Answer: b

1. Which of the following is the default return value of functions in C++?
   1. int b) char c) float d) void Answer: a
2. What happens to a function defined inside a class without any complex operations (like looping, a large number of lines, etc)?
   1. It becomes a virtual function of the class b) It becomes a default calling function of the class

c) It becomes an inline function of the class d) The program gives an error Answer: c

1. What is an inline function?
   1. A function that is expanded at each call during execution b) A function that is called during compile time

c) A function that is not checked for syntax errors d) A function that is not checked for

semantic analysis Answer: a

1. An inline function is expanded during
   1. compile-time b) run-time c) never expanded d) end of the program

Answer: a

1. In which of the following cases inline functions may not word?
2. If the function has static variables.
3. If the function has global and register variables.
4. If the function contains loops
5. If the function is recursive

a) i, iv b) iii, iv c) ii, iii, iv d) i, iii, iv Answer: d

1. When we define the default values for a function?
   1. When a function is defined b) When a function is declared

c) When the scope of the function is over d) When a function is called Answer: b

1. Where should default parameters appear in a function prototype?
   1. To the rightmost side of the parameter list b) To the leftmost side of the parameter list

c) Anywhere inside the parameter list d) Middle of the parameter list Answer: a

1. If an argument from the parameter list of a function is defined constant then
   1. It can be modified inside the function b) It cannot be modified inside the function

c) Error occurs d) Segmentation fault Answer: b

1. Which of the following feature is used in function overloading and function with default argument?
   1. Encapsulation b) Polymorphism c) Abstraction d) Modularity Answer: b
2. What will be the output of the following C++ code? #include<iostream>

using namespace std;

int fun(int x = 0, int y = 0, int z)

{ return (x + y + z); } int main()

{

cout << fun(10); return 0;

}

* 1. 10 b) 0 c) Error d) Segmentation fault Answer: c

1. What will be the output of the following C++ code? #include <iostream>

using namespace std; int fun(int=0, int = 0); int main()

{

cout << fun(5); return 0;

}

int fun(int x, int y) { return (x+y); }

a) -5 b) 0 c) 10 d) 5

Answer: d

1. Which of the following is important in a function?
   1. Return type
   2. Function name
   3. Both return type and function name
   4. The return type, function name and parameter list Answer: c
2. Pick the incorrect statement about inline functions in C++?
   1. They reduce function call overheads
   2. These functions are inserted/substituted at the point of call
   3. Saves overhead of a return call from a function
   4. They are generally very large and complicated function Answer: d
3. Inline functions are avoided when
   1. function contains static variables b) function have recursive calls

c) function have loops **d) all of the mentioned**

Answer: d

1. Pick the correct statement.
   1. Macros and inline functions are same thing
   2. Macros looks like function calls but they are actually not
   3. Inline functions looks like function but they are not
   4. Inline function are always large Answer: b
2. Which functions of a class are called inline functions?
   1. All the functions containing declared inside the class
   2. All functions defined inside or with the inline keyword
   3. All the functions accessing static members of the class
   4. All the functions that are defined outside the class Answer: b
3. The operator used for dereferencing or indirection is a) \* b) & c) -> d) –>> Answer: a
4. Which of the following is a correct identifier in C++?
   1. 7var\_name b) 7VARNAME

c) VAR\_1234 d) $var\_name Answer: c

1. Which of the following is called address operator?

a) \* b) & c) \_ d) % Answer: b

1. Which of the following is used for comments in C++?
   1. // comment b) /\* comment \*/

c) both // comment or /\* comment \*/ d) // comment \*/ Answer: c

1. What are the actual parameters in C++?
   1. Parameters with which functions are called
   2. Parameters which are used in the definition of a function
   3. Variables other than passed parameters in a function
   4. Variables that are never used in the function Answer: a
2. What are the formal parameters in C++?
   1. Parameters with which functions are called
   2. Parameters which are used in the definition of the function
   3. Variables other than passed parameters in a function
   4. Variables that are never used in the function Answer: b
3. Which function is used to read a single character from the console in C++?
   1. cin.get(ch) b) getline(ch) c) read(ch) d) scanf(ch) Answer: a
4. Which function is used to write a single character to console in C++?
   1. cout.put(ch) b) cout.putline(ch) c) write(ch) d) printf(ch) Answer: a
5. What are the escape sequences?
   1. Set of characters that convey special meaning in a program
   2. Set of characters that whose use are avoided in C++ programs
   3. Set of characters that are used in the name of the main function of the program
   4. Set of characters that are avoided in cout statements Answer: a
6. Which of the following escape sequence represents carriage return?
   1. \r b) \n c) \n\r d) \c Answer: a
7. Which of the following escape sequence represents tab?
   1. \t b) \t\r c) \b d) \a Answer: a
8. Out of the following, which is not a member of the class?
   1. Static function b) Friend function c) Constant function d) Virtual function Answer: b
9. What is the other name used for functions inside a class?
   1. Member variables b) Member functions c) Class functions d) Class variables Answer: b
10. Which of the following cannot be a friend?
    1. Function b) Class c) Object d) Operator function Answer: c
11. Which of the following is an abstract data type?
    1. int b) float c) class d) string Answer: c
12. Which of the following is correct?
    1. Friend functions can access public members of a class
    2. Friend functions can access protected members of a class
    3. Friend functions can access private members of a class
    4. All of the mentioned Answer: d
13. What is the difference between delete and delete[] in C++?
    1. delete is used to delete normal objects whereas delete[] is used to pointer objects
    2. delete is a keyword whereas delete[] is an identifier
    3. delete is used to delete single object whereas delete[] is used to multiple(array/pointer of) objects
    4. delete is syntactically correct but delete[] is wrong and hence will give an error if used in any

case Answer: c

1. Which of the following is correct about new and malloc?
2. new is an operator whereas malloc is a function
3. new calls constructor malloc does not
4. new returns required pointer whereas malloc returns void pointer and needs to be typecast

a) i and ii b) ii and iii c) i and iii d) i, ii and iii Answer: d

1. Which of the following loops is best when we know the number of iterations?
   1. while b)do-while c)for d) all the above Answer: c

# UNIT – III

1. Which of the following is not a type of Constructor in C++?
   1. Default constructorb) Parameterized constructor

c) Copy constructor d) Friend constructor Answer: d

1. Which of the following constructors are provided by the C++ compiler if not defined in a class?
   1. Copy constructor b) Default constructor

c) Assignment constructor d) All of the mentioned Answer: d

1. What will be the output of the following C++ program? #include <iostream>

usingnamespacestd; class A{

public:

A(){

}

~A(){

}

};

cout<<"Constructor called\n"; cout<<"Destructor called\n";

int main(intargc, charconst\*argv[])

{

A \*a =new A[5]; delete[] a; return0;

}

* 1. Segmentation fault
  2. “Constructor called” five times and then “Destructor called” five times
  3. “Constructor called” five times and then “Destructor called” once
  4. Error Answer: b

1. What will be the output of the following C++ code? #include <iostream>

using namespace std; main()

{

double a = 21.09399; float b = 10.20;

int c ,d;

c = (int) a;

d = (int) b;

cout << c <<' '<< d; return 0;

}

a) 20 b) 10 c) 21 d) 10

Answer: c

1. What will be the output of the following C++ code? #include<iostream>

using namespace std; class Test

{

protected:

int x; public:

Test (int i):x(i) { }

void fun() const { cout << "fun() const " << endl; } void fun() { cout << "fun() " << endl; }

};

int main()

{

Test t1 (10); const Test t2 (20); t1.fun();

t2.fun(); return 0;

}

* 1. fun() fun() const b)fun() const fun()

c) fun() fun() d) fun() const fun() const Answer: a

1. What does a class in C++ holds?
   1. data b) functions c) both data & functions d) arrays Answer: c
2. How many specifiers are present in access specifiers in class? a) 1 b) 2 c) 3 d) 4

Answer: c

1. Which is used to define the member of a class externally? a) : b) :: c) # d) !!$ Answer: b
2. Which other keywords are also used to declare the class other than class?
   1. struct b) union c) object d) both struct & union Answer: d
3. What will be the output of the following C++ code? #include <iostream>

using namespace std; class rect

{

int x, y; public:

void val (int, int); int area ()

{

return (x \* y);

}

};

void rect::val (int a, int b)

{

x = a; y = b;

}

int main ()

{

rect rect; rect.val (3, 4);

cout << "rect area: " << rect.area(); return 0;

}

* 1. rect area: 24 b) rect area: 12 c) compile error d) rect area: 56

Answer: b

1. What will be the output of the following C++ code? #include <iostream>

using namespace std; class CDummy

{

public:

int isitme (CDummy& param);

};

int CDummy::isitme (CDummy& param)

{

if (&param == this) return true;

else

return false;

}

int main ()

{

CDummy a; CDummy \*b = &a; if (b->isitme(a))

{

cout << "execute";

}

else

{

cout<<"not execute";

}

return 0;

}

* 1. execute b) not execute c) error d) both execute & not execute

Answer: a

1. Which of the following is a valid class declaration?
   1. class A { int x; }; b) class B { } c) public class A { } d) object A { int x; }; Answer: a
2. The data members and functions of a class in C++ are by default
   1. protected b) private c) public d) public & protected Answer: b
3. Constructors are used to
   1. initialize the objects b) construct the data members

c) both initialize the objects & construct the data members d) delete the objects Answer: a

1. When struct is used instead of the keyword class means, what will happen in the program?
   1. access is public by default b) access is private by default

c) access is protected by default d) access is denied Answer: a

1. Which category of data type a class belongs to?
   1. Fundamental data type b) Derived data type

c) User defined derived data type d) Atomic data type Answer: c

1. Which operator a pointer object of a class uses to access its data members and member functions?

a) . b) -> c) : d) ::

Answer: b

1. What is the correct syntax of accessing a static member of a Class? class A

{

public:

}

static int value;

* 1. A.value b) A::value c) A->value d) A^value Answer: b

1. How the objects are self-referenced in a member function of that class.
   1. Using a special keyword object b) Using this pointer

c) Using \* with the name of that object d) By passing self as a parameter in the member function

Answer: b

1. What does a mutable member of a class mean?
   1. A member that can never be changed
   2. A member that can be updated only if it not a member of constant object
   3. A member that can be updated even if it a member of constant object
   4. A member that is global throughout the class Answer: c
2. What is the role of a constructor in classes?
   1. To modify the data whenever required
   2. To destroy an object
   3. To initialize the data members of an object when it is created
   4. To call private functions from the outer world Answer: c
3. Why constructors are efficient instead of a function init() defined by the user to initialize the data members of an object?
   1. Because user may forget to call init() using that object leading segmentation fault
   2. Because user may call init() more than once which leads to overwriting values
   3. Because user may forget to define init() function
   4. All of the mentioned Answer: d
4. What is a copy constructor?
   1. A constructor that allows a user to move data from one object to another
   2. **A constructor to initialize an object with the values of another object**
   3. A constructor to check the whether to objects are equal or not
   4. A constructor to kill other copies of a given object. Answer: b
5. In the following C++ code how many times the string “A’s constructor called” will be printed? #include <iostream>

#include <string> using namespace std; class A{

int a;

public:

};

class B{

A(){

}

cout<<"A's constructor called";

public:

static A a;

B(){

cout<<"B's constructor called";

}

static A get(){

return a;

}

};

A B::a;

int main(int argc, char const \*argv[])

{

B b;

A a1 = b.get();

A a2 = b.get();

A a3 = b.get();

}

a) 3 b) 4 c) 2 d) 1 Answer: d

1. What happens if a user forgets to define a constructor inside a class?
   1. Error occurs b) Segmentation fault

c) Objects are not created properly d) Compiler provides a default constructor to avoid faults/errors

Answer: d

1. How many parameters does a default constructor require? a) 1 b) 2 c) 0 d) 3

Answer: c

1. How constructors are different from other member functions of the class?
   1. Constructor has the same name as the class itself
   2. Constructors do not return anything
   3. Constructors are automatically called when an object is created
   4. All of the mentioned Answer: d
2. How many types of constructors are there in C++? a) 1 b) 2 c) 3 d) 4 Answer: c
3. What will be the output of the following C++ code? #include <iostream>

#include <string> using namespace std; class A{

mutable int a; public:

A(){

cout<<"Default constructor called\n";

}

A(const A& a){

cout<<"Copy Constructor called\n";

}

};

int main(int argc, char const \*argv[])

{

A obj;

A a1 = obj;

A a2(obj);

}

* 1. Default constructor called, Copy Constructor called
  2. Default constructor called, Copy Constructor called, Copy Constructor called
  3. Default constructor called, Default constructor called, Copy Constructor called
  4. Copy Constructor called, Default constructor called, Copy Constructor called Answer: b

1. What will be the output of the following C++ code? #include <iostream>

#include <string> using namespace std; class A{

mutable int a;

public:

A(){

}

cout<<"A's default constructor called\n";

A(const A& a){

cout<<"A's copy Constructor called\n";

};

class B{ public:

};

}

A obj; B(){

}

cout<<"B's Constructor called\n";

int main(int argc, char const \*argv[])

{

B b1; B b2;

}

* 1. B's Constructor called, B's Constructor called
  2. B's Constructor called, A's default constructor called, B's Constructor called, A's default constructor called
  3. A's default constructor called, B's Constructor called, A's default constructor called, B's Constructor called
  4. A's default constructor called, B's Constructor called, A's copy Constructor called Answer: c

1. What is the role of destructors in Classes?
   1. To modify the data whenever required
   2. To destroy an object when the lifetime of an object ends
   3. To initialize the data members of an object when it is created
   4. To call private functions from the outer world Answer: b
2. What is syntax of defining a destructor of class A?

a) A(){} b) ~A(){} c) A::A(){} d) ~A(){};

Answer: b

1. When destructors are called?
   1. When a program ends b) When a function ends

c) When a delete operator is used d) All of the mentioned Answer: d

1. What is Inheritance in C++?
   1. Wrapping of data into a single class b) Deriving new classes from existing classes

c) Overloading of classes d) Classes with same names Answer: b

1. How many specifiers are used to derive a class? a) 1 b) 2 c) 3 d) 4 Answer: c
2. Which of the following class allows to declare only one object of it?
   1. Abstract class b) Virtual class c) Singleton class d) Friend class Answer: c
3. Which of the following is not a type of Constructor?
   1. Friend constructor b) Copy constructor c) Default constructor d) Parameterized constructor Answer: a
4. Which of the following provides a programmer with the facility of using object of a class inside other classes?
   1. Inheritance **b) Composition** c) Abstraction d) Encapsulation Answer: b
5. Which operator is overloaded for a cout object? a) >> b) << c) < d) > Answer: b
6. Which of the following cannot be used with the virtual keyword?
   1. Class b) Member functions c) Constructors d) Destructors Answer: c
7. Which of the following is correct?
   1. C++ allows static type checking
   2. C++ allows dynamic type checking.
   3. C++ allows static member function to be of type const.
   4. C++ allows both static and dynamic type checking Answer: d
8. What happens if a class does not have a name?
   1. It will not have a constructor b) It will not have a destructor

c) It is not allowed d) It will neither have a constructor or destructor Answer: b

1. Which of the following is correct in C++?
   1. Classes cannot have protected data members b) Structures can have member functions

c) Class members are public by default d) Structure members are private by default

Answer: b

1. Which of the following is used to make an abstract class?
   1. By using virtual keyword in front of a class declaration
   2. By using an abstract keyword in front of a class declaration
   3. By declaring a virtual function in a class
   4. By declaring a pure virtual function in a class Answer: d
2. Which of the following is correct?
   1. A class is an instance of its objects
   2. **An object is an instance of its class**
   3. A class is an instance of the data type that the class have
   4. An object is an instance of the data type of the class Answer: b
3. Which of the following is correct about new and malloc?
   1. Both are available in C
   2. Pointer object initialization of a class with both new and malloc calls the constructor of that class
   3. Pointer object initialization of a class using new involves constructor call whereas using malloc does not involve constructor call
   4. Pointer object initialization of a class using malloc involves constructor call whereas using new does not involve constructor call

Answer: c

1. What will be the output of the following C++ code? #include <iostream>

using namespace std; class A{

public:

A(){

}

~A(){

}

};

cout<<"Constructor called\n"; cout<<"Destructor called\n";

int main(int argc, char const \*argv[])

{

A \*a = new A[5]; delete a;

return 0;

}

* 1. “Constructor called” five times and then “Destructor called” five times
  2. “Constructor called” five times and then “Destructor called” once
  3. Error
  4. Segmentation fault Answer: d

1. What will be the output of the following C++ code? #include <iostream>

using namespace std; class A

{

int a;

A() { a = 5;}

};

int main()

{

A \*obj = new A; cout << obj->a;

}

* 1. 5 b) Garbage value c) Compile-time error d) Run-time error Answer: c

1. What happens if the following C++ statement is compiled and executed? int \*ptr = NULL;

delete ptr;

* 1. The program compiled successfully but throws an error during run-time
  2. The program gives a compile-time error
  3. The program is not semantically correct
  4. The program is compiled and executed successfully Answer: d

1. Identify the incorrect constructor type.
   1. Friend Constructor b)default Constructor c)copy constructor d)parameterized constructor Answer: a
2. Which one of the following operators could be overloaded?
   1. - b)sizeof c):: d).\* Answer: a
3. Which of the following could not be used with friend?

a) = b)() c)[] d)All the above Answer: d

1. Type conversion may happened
   1. basic to class type b) class to basic

c) one class to another d) all the above Answer: d

1. Operator overloading is called as
   1. compile time polymorphism b)run time polymorphism c)all the above d) none of the above

Answer: a

1. The compiler does not support automatic type conversion for
   1. user defined data type b)predefined c)all the above d)none

Answer: a

1. Which of the followings is/are automatically added to every class, if we do not write our own.
   1. Copy Constructor b) Assignment Operator

c) A constructor without any parameter d) All of the above Answer: d

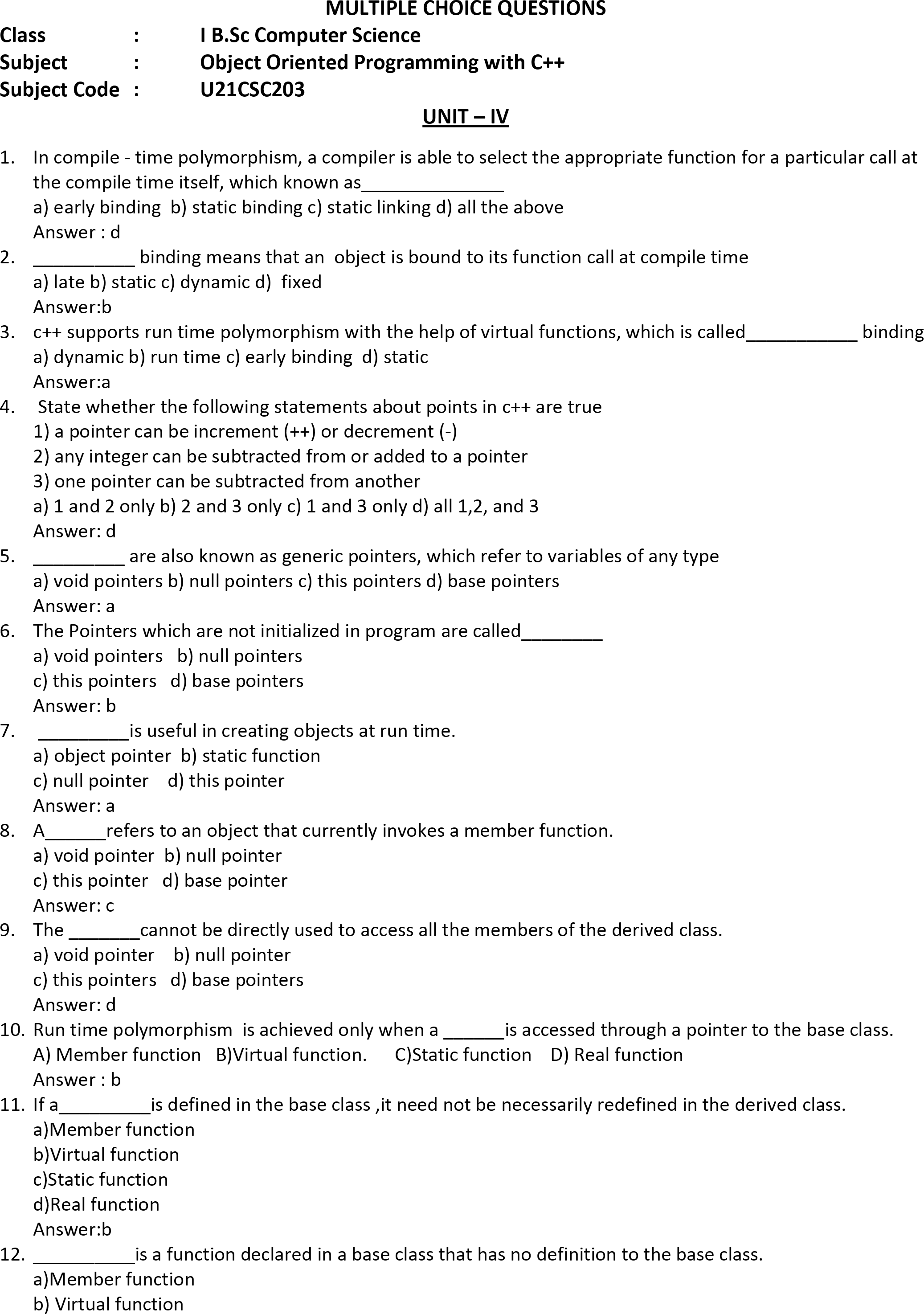
1. Which of the following gets called when an object is being created?
   1. Constuctor b)Virtual Function c)Destructors d)Main Answer: a
2. Destructor has a same name as the constructor and it is preceded by?

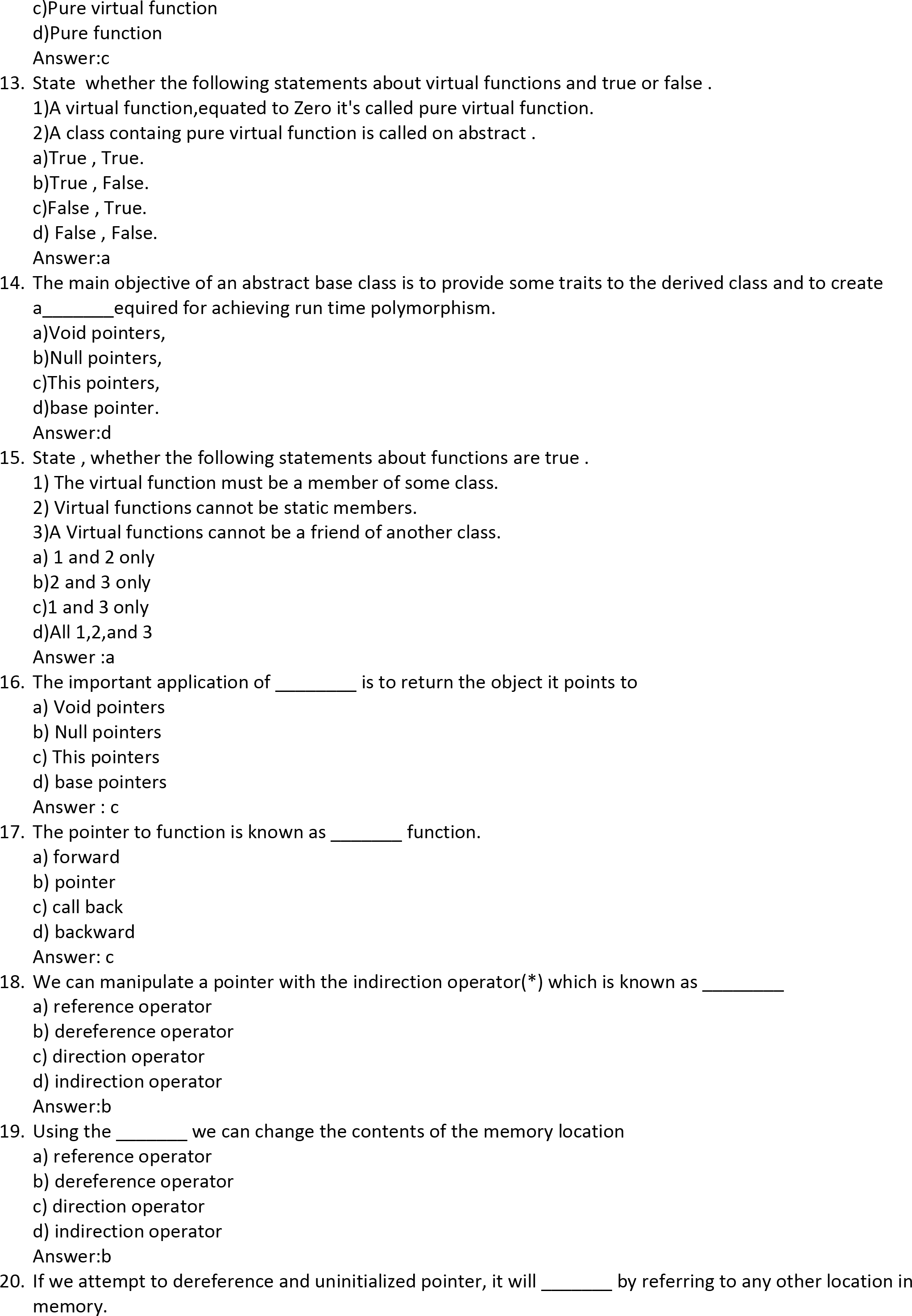
a) ! b) ? c) ~ d) $ Answer: c

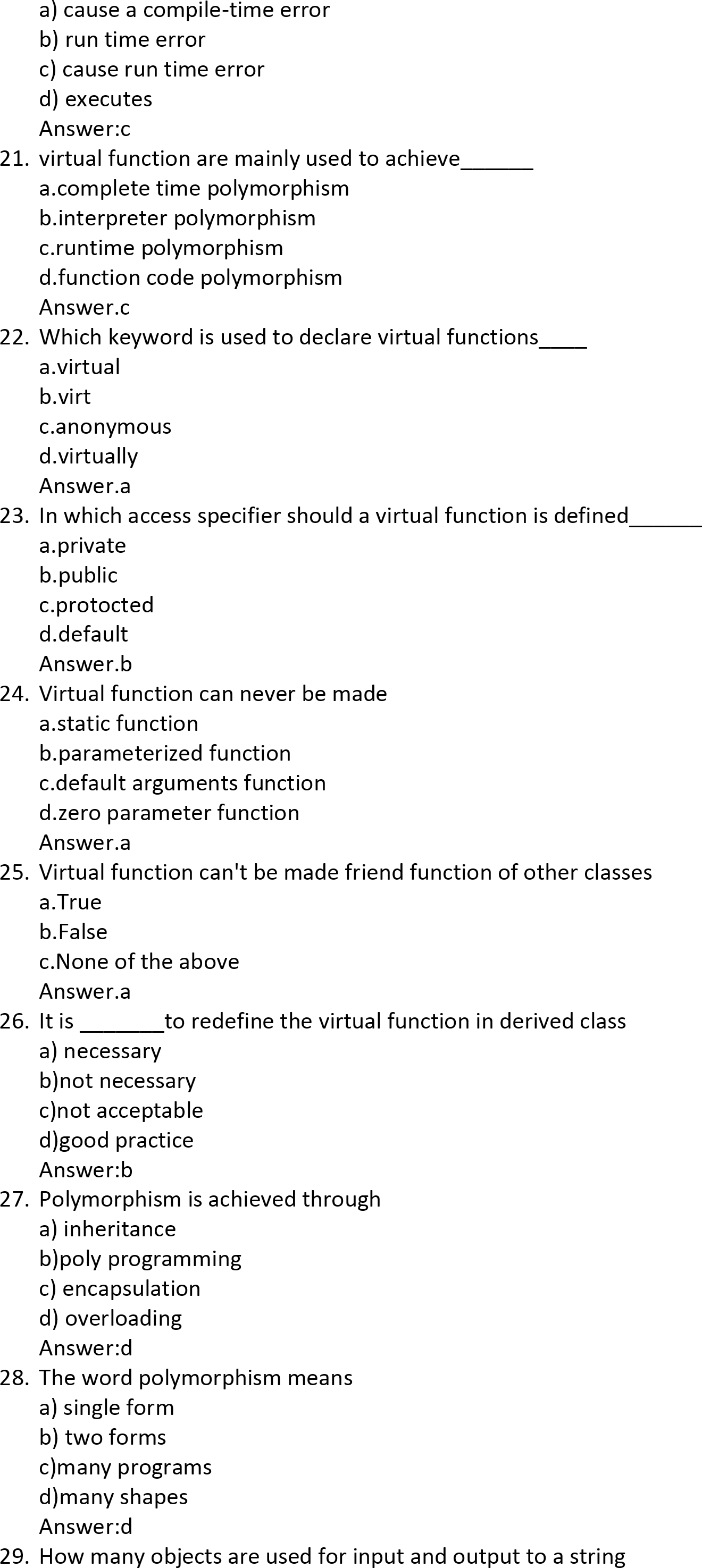
Like constructors, can there be more than one destructors in a class?

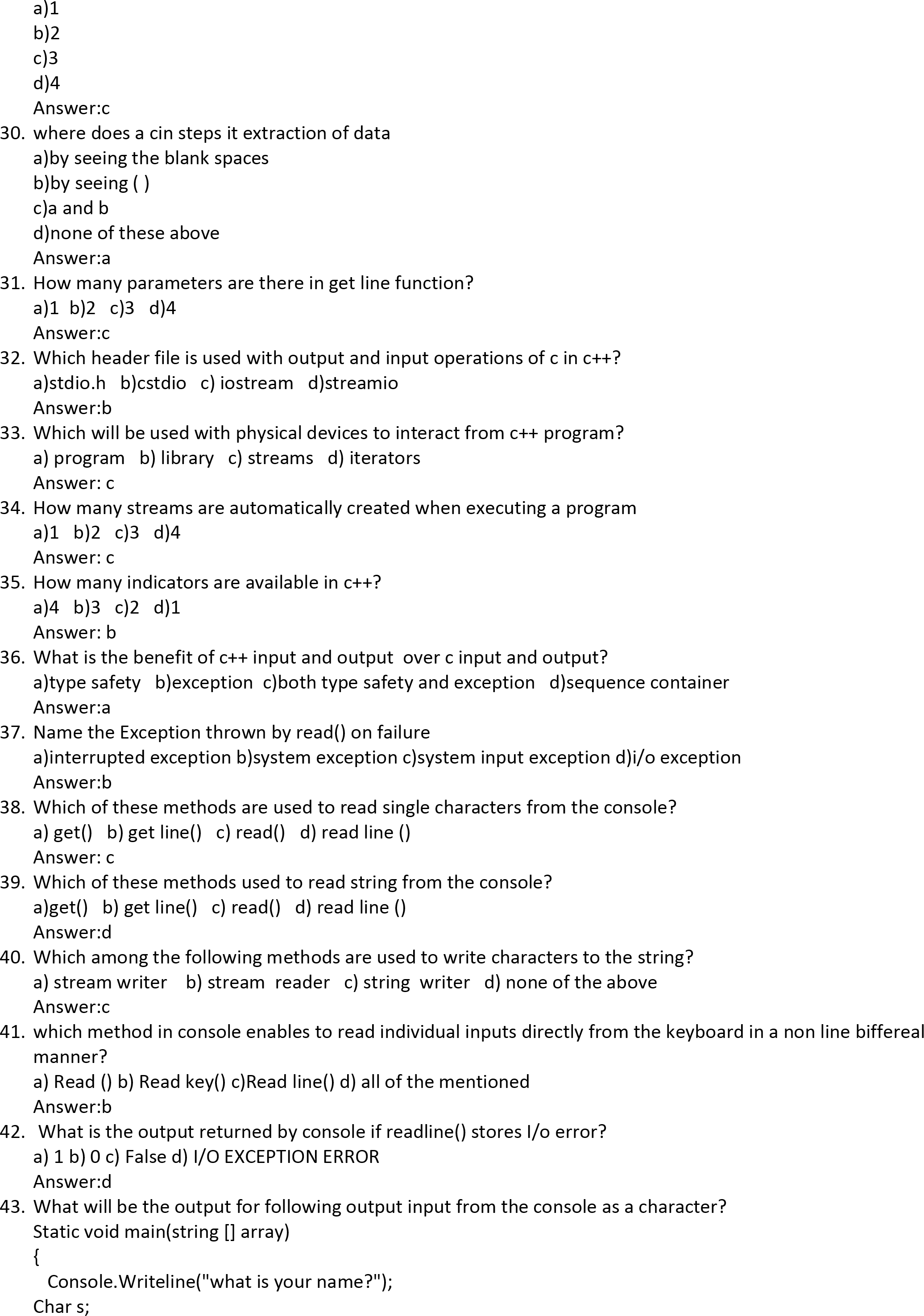
* 1. Yes b) No c) May Be d) Can't Say Answer: b

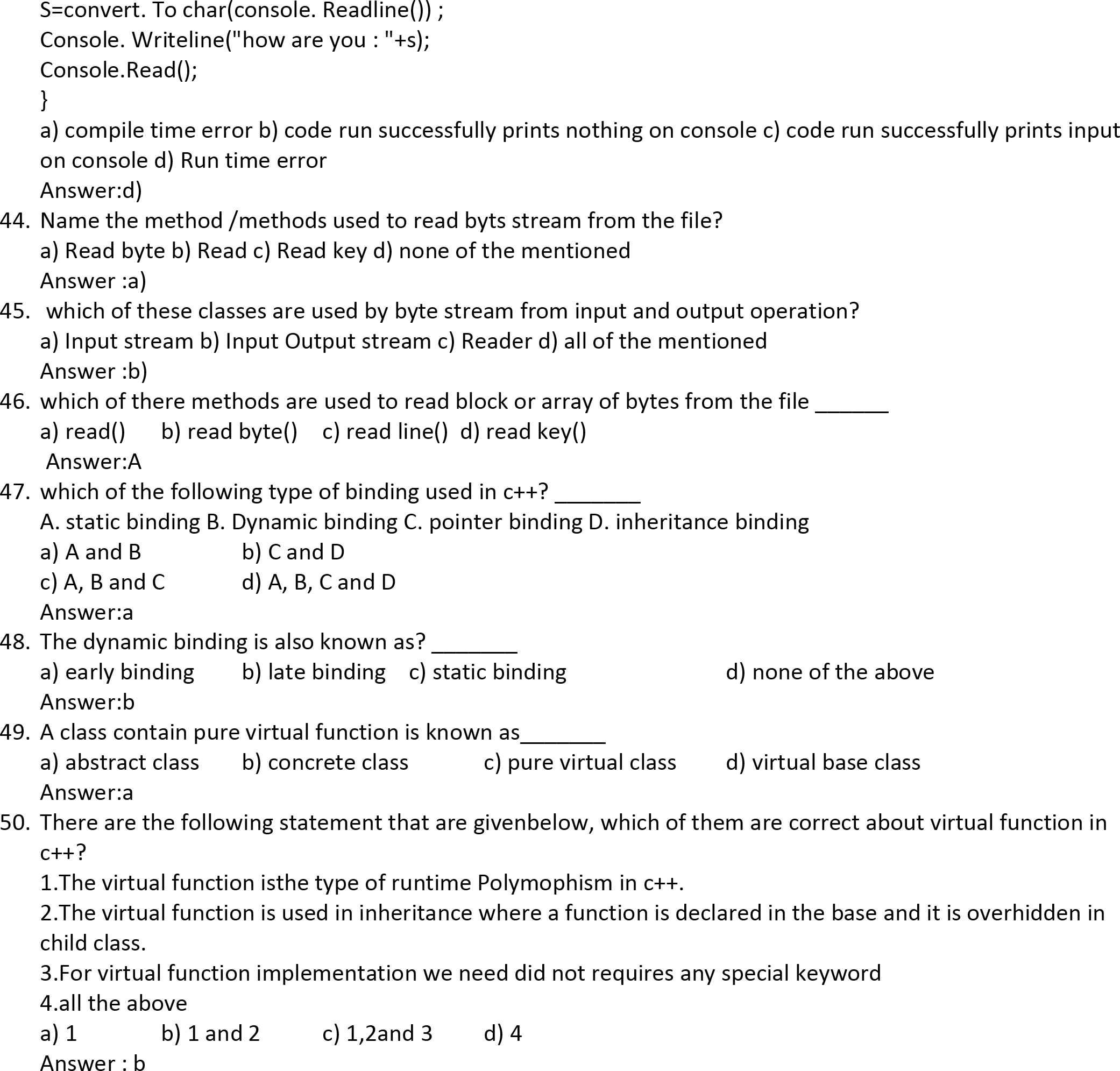
1. State whether the following statements about the constructor are True or False. i) constructors should be declared in the private section. ii) constructors are invoked automatically when the objects are created.
   1. True,True b) True,False c) False,True d) False,False Answer: c
2. Which contructor function is designed to copy object of same class type?
   1. **Copy constructor** b) Create constructor c)Object constructor d) Dynamic constructor Answer: a











51)Most important amongst these it that is premisets \_\_\_\_\_\_\_

a)code reusability b)code reusing c)exiting code d)code readability

Answe:a

52)\_\_\_\_\_\_\_\_\_\_\_save time and money and increasing a program realibility

a)reusing existing code b)resusing entiring code c)exiting code d)none of these

Answer:a

53.\_\_\_\_\_\_\_\_\_\_can be also help in the orginal conceptualization of a programming problems

a)class b)object c)dynamic binding d)inheritance

Answer:d

54)A programming can use a \_\_\_\_\_\_\_\_\_\_\_\_created by another person or company and without un modifying it

a)object b)class c)message passing d)inheritance

Answer:b

55.The \_\_\_\_\_\_\_\_\_class can only increment the counter are not decrease it

a)text b)index c)intest d)debug

Answer:b

56)The index class works well and has been througtly \_\_\_\_\_\_\_\_\_\_\_\_\_

a)testes b)untested c)tested and debugged d)debugged

Answer:c

57)We might not have access to its source code especially if it had distributed as the part of a\_\_\_\_\_\_\_\_\_\_\_

a)class b)object c)dynamic binding d)massage passing

Answer:a

58)A member declared as \_\_\_\_\_\_\_\_which serve limited purpose inheritance

a)public b)protected c)private d)processed

Answer:b

59)A class can inherit the attributes of two are more classes is known a

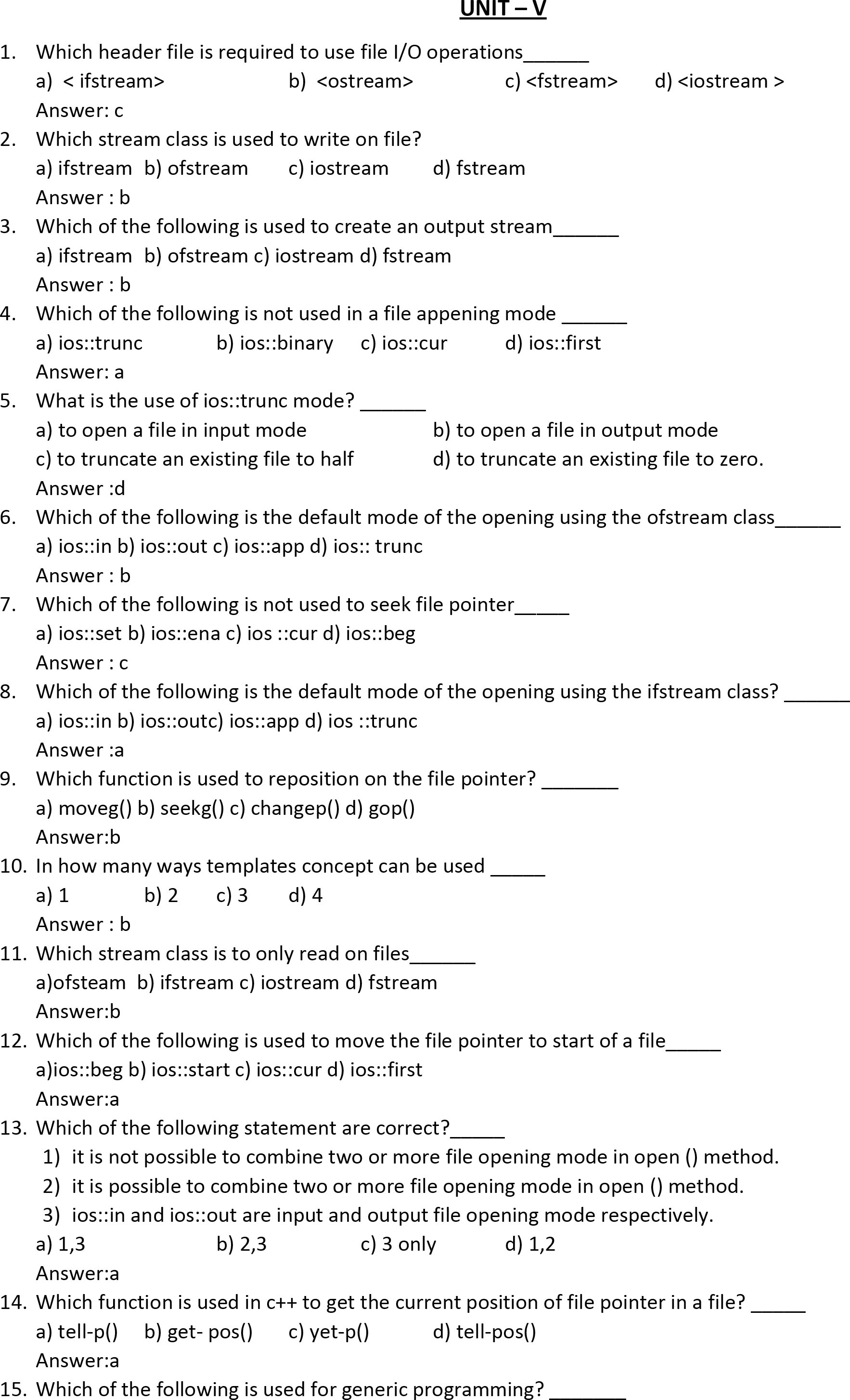
a)mulitiple inheritance b)multilevel inheritance c)inheritance d)single inheritance

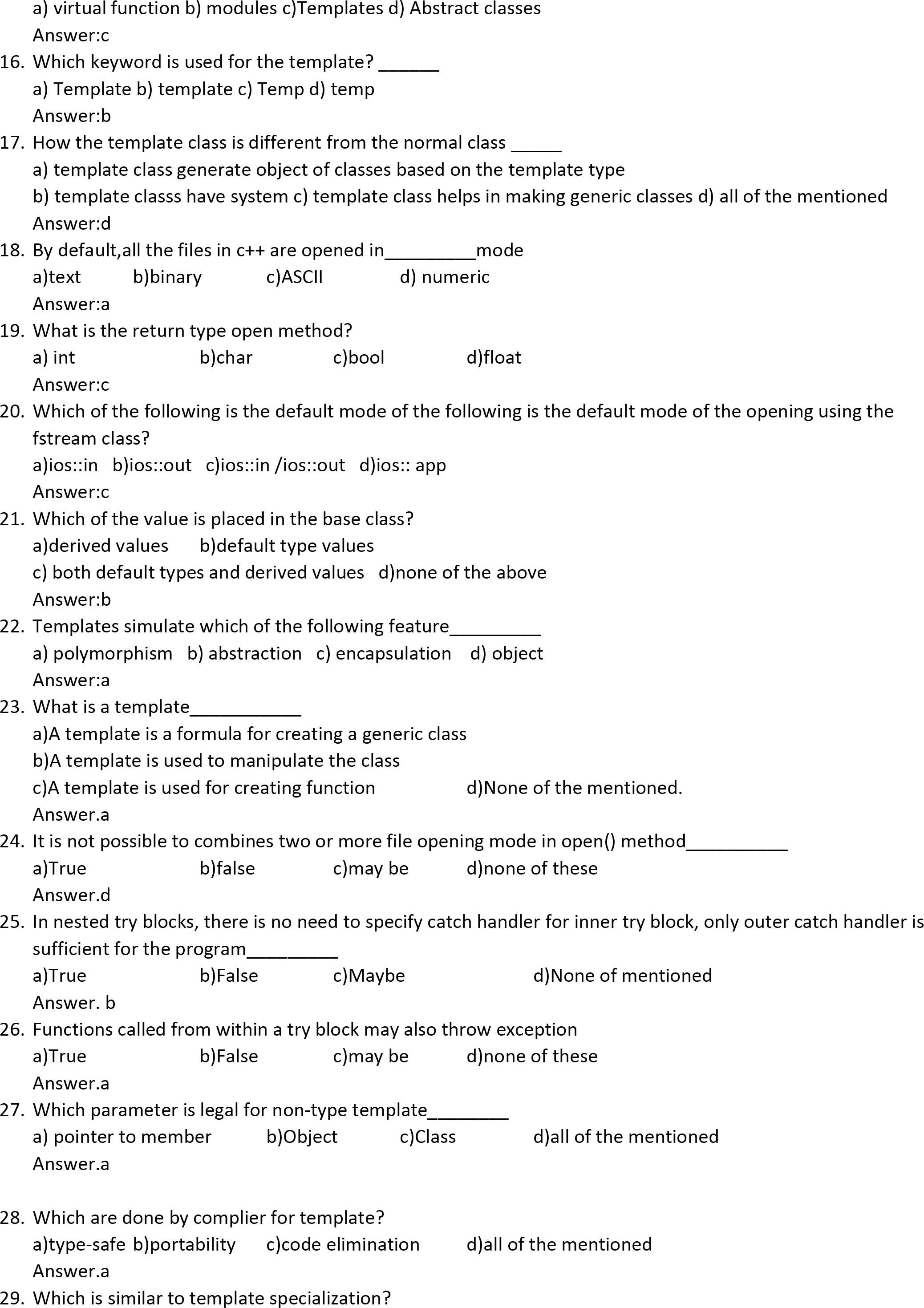
Answer:a

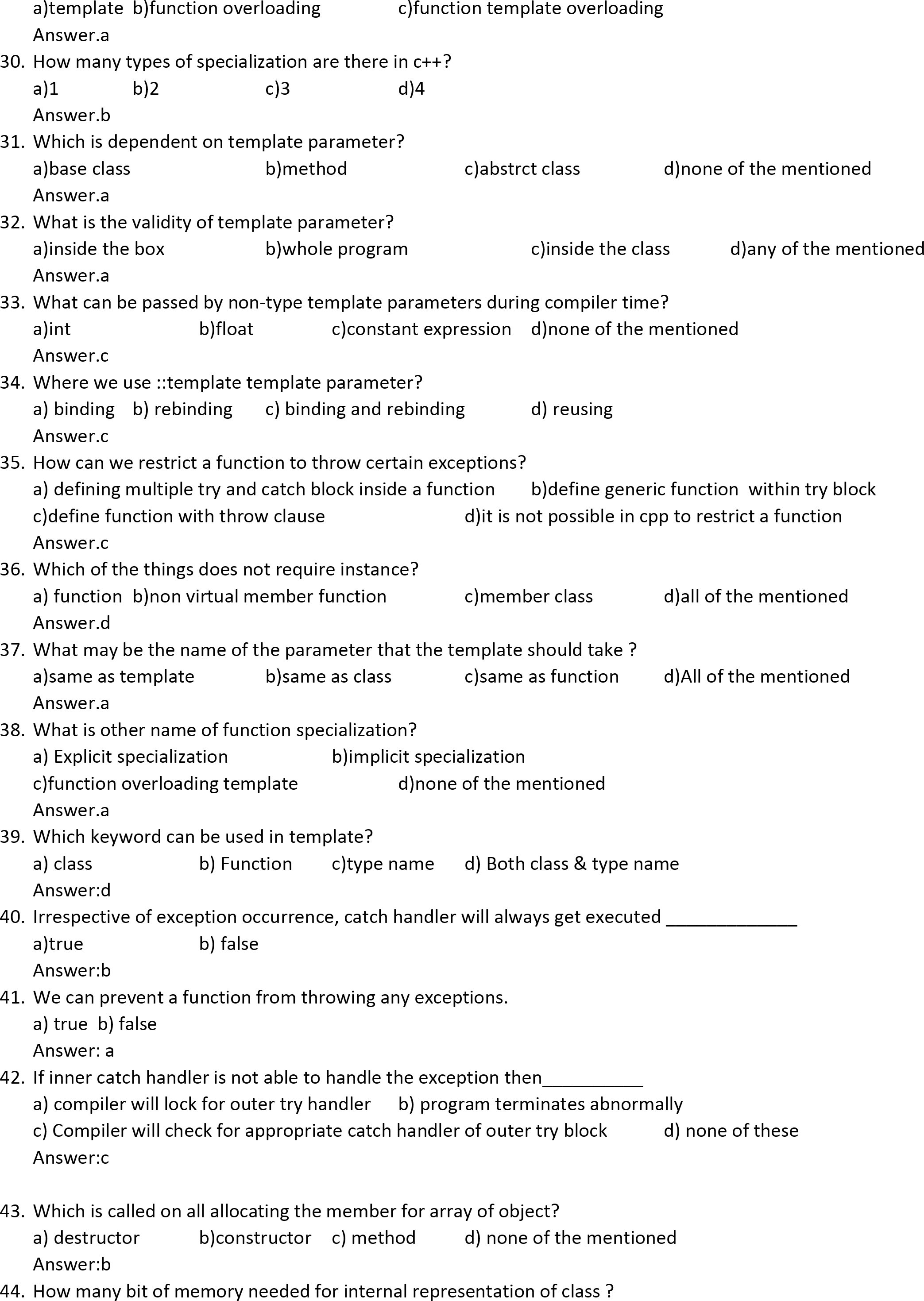
60)In c++ such problems can be converted into a class\_\_\_\_\_\_\_

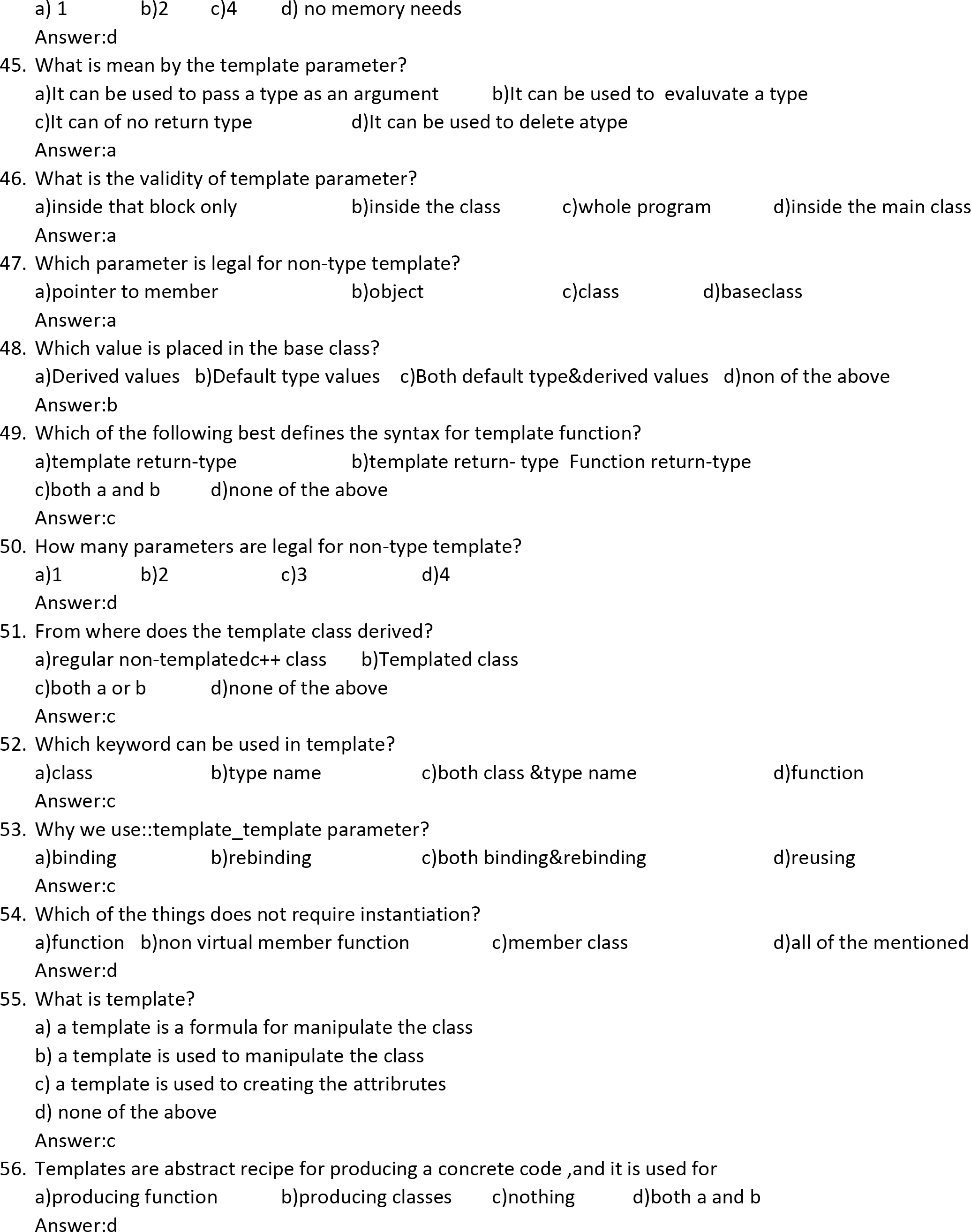
a)class b)object c)inheritance **d)hierarchices**

Answer:d









57. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_are a special functions that can be included in the I/O statements

a)manipulators b)operator c)truncates d)functions

Answer:a

58. Floating point numbers can be done by using the\_\_\_\_\_\_\_\_\_\_\_\_

a)manipulators b)precision c)console d)write()

Answer:b

59. The\_\_\_\_\_\_\_\_\_\_of a field are filled with white space by default

a)unused b)reused c)used d)none of this

Answer:a

60. The flag set by \_\_\_\_\_\_\_\_\_\_remain effective until they are reset

**a)self()**  b)width() c)self d)none of these

Answer:a