# Doc\*\*\*\*\*

1. Which of the following will be the output of the C++ program below? int main()

{

int Age(4) ; Age[0) = 5;

Age[1] = 1;

Age [2] = 21;

Age[3] = 12;

for int person 0; person<4; person

if ( (Age (person) - 1) || (Age(person] = 2))

Cout<< "Baby

if (Age [person) > 21)

cout << "Adult

if ( (Age (person] > 12) 66 (Age(person] < 21)).

if((age(person) cout << "Teen else

pout "Youngster (Agelperson)

< 21))

return 1.

* 1. Youngster Youngster Youngster Youngster
  2. Youngster Baby Aduit Teen
  3. Baby Youngster Baby Youngster Baby Youngster Baby Youngster 4)O Baby Baby Baby Baby

- Teen Adult Baby Youngster Ans) 3

1. Which of the following will be the output of executing the ANSI C program below? #include <stdio.h>

int main(void) int x= 4, y,z;

Y= --x;

Z = x--;

Printf(“%d %d %d”, x,y,z);

Return 0;

1) 233

2) 332

3) 432

3) 223

4) 433

Ans) 1

1. Which of the following C++ algorithms do NOT modify the elements on which they operate?
2. std:: for each
3. std::all of
4. std::fill

.4) std::generate

1. std::transform Ans) 12

4) Which of the following ANSI C statements contain correct printf() conversations to generate the output below?

31.240000 6.360000 5.460000

* 1. printf ("%Lf %f %lf “, a, b, c) :
  2. printf (%f %lf Lf%” a,b,c,);
  3. printf (%lf %lf %f” a,b,c)
  4. printf(%f %f %f” , a,b,c);
  5. printf(“%Lf %Lf %Lf”, a , b, c);

ans) 2

1. Which of the following statements correctly describe the ANSI C program below? # include <stdio.h>

Int main(void)

{

Float a = 12.25, b= 13.65;

If( a = b)

print(("a and b are equal"); else

printf("a and b are not equal\*); return 0.

statements correctly describe the ANSI C program below

1. The value of a is compared to the value of b.
2. The statement if (a = b) will report a compilation error.
3. The output of the program will be "a and b are equal'.
4. a is assigned the value of b.
5. The code in the else clause will never be reached.

Ansr) 3,4,5

1. Which of the following statements correctly describe the \*ad? command?
   1. By default, it will display 32 byes of input data on each output
   2. It will only work correctly for files that have fixed length record
   3. If it is executed with the ' option and the file is not an ASCI
   4. option ‘-x’ will convert the input into hexadecimal.
   5. It can only be used on ASCII files. Answr) 4
2. Which of the following is a c out put

{

1,2,3,4

4,3,2,1

7,8,9,0

};

Printf (“%p %p\n”,sa+1 ,&a+1);

Return 0;

}

i) 2 3

ii) 64016 64048

iii) 64004 64016

iv) 64001 64001

v) 64004 64004

Answr) 5

1. Int amount( int ii, int jj)

{

Int kk, ll;

Kk = ii + jj; Ll = ii \* jj;

Return (kk , ll) ;

}

i) 12 7

ii) 12 12

iii) 3 4

Ansr) 2

1. A user opens a 100 line file for editing by issuing the \*vi my filet command and moves the cursor to the 40th column of the 70th

line. The editor is not in INSERT mode. Which of the following statements are correct?

Answr) 1,2,3

i)

The "dfa' command will delete from the current cursor position up to and including the first 'a' on the current line.

ii)

The 3D' command will delete the rest of the line to the right of the cursor and the will delete the following 2 lines.

iii)

The '$' command will delete the entire contents of the current line and put the cursor at the beginning of the line. The editor wil

be in INSERT mode. iv)

The '$' command will move the cursor to the end of the file v)

The 'dta' command will delete from the current cursor position up to and including the first 'a' on the current line

1. Which of the following statements correctly describe UNIX commands?
   1. The 'whatis' command can be used to determine the terminal device and login of a user
   2. The 'whatis' command can be used to determine the full file specification of a command.
   3. The 'apropos' command displays which 'man pages' to read for a given keyword.
   4. The 'which' command can be used to determine the full file specification of a command.

-5) The 'which' command can be used to determine the terminal device and login of a user,

Answr) 2,3,5

11)

A user opens an existing file for editing by issuing the “vi my\_file” command and

moves the cunsor to the 70th

not in INSERT mode. Which of thefollowing statements correcuy

-desenbe the resile at ovariw

1. successful, the cursor will move to the line belore the line containing the search sinne
2. It will search backwards (from the line above the current line) for a line that begins with a and ends i
3. It will search backwards (from the line above the current line) for a word that begins with "a' and end
4. It will search backwards (from the curt

nt cursor position) for a word that begins with a and ends in

1. If the search is successful. the cursor will move to the line containing the search string

Answr) 2 3 4

//4

12)

user opens an existing file for editing by issuing the \*vi my file' command and has not performed any editing actions. Which of

the following statements are correct?

* + 1. The :1, 2w my \_old file command will always save the first 2 lines in fle my \_old\_ file.
    2. The :w S.zz command will attempt to copy my\_file to file my\_file.22. iii )The : command will cause the editor to reread the file.

The 2z command will cause the editor to be exited

iv) The iws command will save the file currently being edited. Answer 123

13)

A user opens an existing file for editing by issuing the \*vi The editor is not in INSERT mode. Which of the follo wing stalements are correct?

land and moves the

cursor to the 50th line of the 100 lir i)

The 3li command moves the cursolup 3 lines, movesthe cursor to the beginnling of the line, and puts the eddor in INSERImode.

ii)

The 3jo command moves the cursor up 3 lines, creates a blank line, and puts the editor in INSERT mode

iii)

The 31o command moves the cursor up 3 lines, creates & olank line, ind puts the editor in INSERT mada

iv)

The 3ki command moves the cursor up 3 lines and puts the editor in INSERT mode v)

The 3hi command moves the cursor up 3 lines, moves the cursor to the beginning of the line, and puts t

Answr) 4

14) A user opens an existing fle for editing by issuing the \*Vi any file' command. The 701current line. The editor is not in INSERTline is marked as a and the

mode. Which of the following i)

The "by" a "bp command will append 3 lines to buffer "b' and only insert those 3 lines immediately after the line where the

Is currently located ii)

The "by”a"bp command will append 2 lines to buffer “b”and only insert those 2 lines immediately after the line where the cursor is currently located.

iii)

The”by”a command will append 2 lines to buffer “b”

iv)

The ‘by' ap command will append 3 lines to buffer “b” and only insert those 3 lines

immediately after the line where the cursol is currently located.

v)

The "by" ap command will append 2 lines to buffer “b" and only insert those 2 lines immediately after the line where the cursor

Answr) 124

15)

A user opens an existing file for editing by issuing the \*vi my file' command and moves the cursor to the middle

The editor is not in INSERT mode. Which of the following staterents correctly describe the results of executing a

:., +33/(ul)yp/los/g command?

1. It will change all occurrences of YP' or yp' in the current line and the next 3 lines to 'los'
2. It will change all occurrences of 'Yp' or 'yp' in the current line and the next 3 lines to "los'
3. It will change all occurrences of 'uyp' or \*lyp' in the current line and the next 2 lines to 'los'.

iv ) It will change all occurrences of 'uyp' or 'lyp' in the current line and the next 3 lines to "los'

v) It will change all occurrences of 'YP' or 'yp' in the current line and the next 2 lines to 'los',

Answr) 124

//4

1. Box rhs(v);

Return volume ==rhs.volume;

* 1. The existing overloaded operators already handle the third comparison. ii)

Implement a global function called operator

-which takes a double and a Box and have it call an existina me operatoraa implementation with conforming parameters.

* + 1. Modify one of the existing member operator== implementations so that it takes an additional parameter.
    2. Implement a global function called operator-

- Which takes a double and a Box and makes it a friend of

v) implement a member operator=- taking a Box and a double Answr) 4

1. return 0?

foo (int as, float bb) return (float) a bha

* 1. float foo(int, int);
  2. int foo(int, float) :
  3. float foo(float, int);
  4. int foo(float, float) ;

-

* 1. float foo(float,float); Answr) 2

1. Which of the following statements about the ANSI C bitwise OR operator (1) are valid?
   1. It can be used to set multiple bits of an integer.
   2. It can be used to multiply an integer by powers of 2 (two).
   3. It can be used to set a bit of an integer.
   4. If either corresponding input bit is 1 (one), the corresponding output bit is 0 (zero).
   5. It can be used to divide an integer by powers of 2 (two). Answr 134
2. if a base class in Ct overloads the assignment operator, which of the following steps need to be taken so that the operat

implemented in the derived classes?

* 1. Declare the overloaded operator in the base class to be virtual.
  2. Implement the overloaded operator as a friend to the base class.
  3. Declare the overloaded operator in the base class to be dynaric.
  4. None. An operator overloaded in a base class is sufficient for all derived classes.
  5. Override the operator in the derived class to account for any changes to the data members of the class.

Answr ) 1,5

1. Which of the following statements correctly describe the usage of operator overloading in C++?
   1. If a class overloads any operators, then all operators of the class must also be overloaded

-ii) An operator may not be overloaded more than once for a given class.

1. It is possible to overload operators for integers.
2. An overloaded operator may be implemented as a member function. v)Any operator may be overloaded.

Answr :2,3,4

1. When overloading C\*\* unary operators, which of the following are legal options with respect to the number of parameters to be used?
   1. Any number of parametens, when the operalor tuntion is a tree standing huhction (nol a dass member).
   2. One (dummy) parameter, when the operator is a particular type of increment/decrement class member function.
   3. No parameters, when the operator function is a class member.
   4. No parameters, when the operator function is a free standing function (not a class member).
   5. One parameter, when the operator function is a free standing function (not a class member).

ANSWR) 2,3,5

1. which of the following scenarios must a C++ destructor be declared virtual in a base class?
   1. When the default constructor of the base class is declared as virtual.
   2. When an implementation for the derived class destructor is not desired.
   3. When the developer wants to ensure that a derived class destructor is invoked when an instanc

class pointer.

* 1. When a derived class allocates system resources which are released in its destructor.
  2. When the destructor of the base class will be doing the clean-up of the derived class data

answr: 3,4

1. A requirements specification for a C\*\* class called Rotator calls for the exclusive management of a critical resour

decided to enforce the exclusiveness, in part, by preventing the class from being copied. In addition, the design call

compiler generated default constructor for this class. Which of the following are steps that may be taken to impleme

requirements? i)

Declare a copy assignment operator, then explicitly delete it such as void operator=(Rotator

1. Declare the compiler generated-constructor to be explicitly defaulted such as Rotator () = defa
2. Let the compiler generate the default constructor and manually declare that it is final such as Rota
3. Let the compiler generate the default constructor and don't make any other reference to it in the cla
4. Declare a copy constructor, then explicitly delete it such as Rotator (Rotator const &)

Answr 125

1. The c++ code below is an example of which of the following types of classes? Class poygon{

Protected :

Int width, height; Public:

Void set\_values(int a, int b) ( palllets=a; packets = b;) Virtual int paliets() =0;

};

* 1. Abstract base class ii)Instantiated class iii)Base class iv)Virtual class v)Private class answr:1,3

void func3 () (

cout <<

\* func3 func4 0 ; void func40)

exception e("func4 throw 10;

m);

i) func3 10 two ii)func3 10 2

1. func3 5 2
2. func3 5 func1 1
3. func3 5 two answr: 3

Which of the following statements correctly describe execution of the highlighted

$grep ^root/etc/passwd Root:qged2627tvxq:0:0:0000=admin(0000),,0258,:/:bin/tcsh

$su

1. The user will be prompted for a password and then /bin/tash will be executed. The current directory wil not
2. The user will be prompted for a password and then /bin/sh will be executed iii)The current directory will be changed to "/\*
3. The system will execute /bin/sh. The current directory will not be changed.
4. The environment will be the same as though root (super user) had logged onto the system.

Answr 135

* + Which of the following declarations are valid ways of defining a template alias in C++?
* typedef template<class T> std: :unique\_ptr< T, std: : function< void (T.) >> Up;
* typedef std : : unique\_ptr< T, std : : function< void (T' ) > > UP;
* typename UP = std: :unique\_ptr< T, std: : function< void (T") >>;
* using up = std: :unique\_ptr< T, std: : function< void (T") >>;
* template<class T> using Up = std: :unique\_ptr< T, std: : function< void (?") >>; Answer: 1 5

// 1

* + Fot the C++ template definition below, which of the following are valid ways to instantiate this template?

Template<class T>;

Class tClass { Public:

T data;

};

* tClass i<int>;
* tclass i (int) ;
* tClass (int) i;
* <int>tclass i;
* tClass<int> i;

--

Answer: 5

* + Question-
* Type of new\_arg is const Type of new arg is not const Type of new arg is not const
* Type of new\_arg is not const Type of new\_arg is not const Type of new arg is const
* Type of new arg is const Type of new arg is const Type of new arg is const
* Type of new\_arg is not const Type of new arg is const Type of new arg is const
* Type of new\_arg is not const Type of new\_arg is not cone Type of new arg is not const

-

Answer: 5

* + Which of the following are valid instantiations of the C++ function template declaration shown below?

template Sclass T, class. .. Argus

vold runt const zo first, count argus. . reat 17

* fun ( 1, 2, 3, 4, 5 );
* fun ( "silly string" );
* fun ( ( 1 , 2, 3, 4, 5 ) );
* fun ( 1, 2.0, "buckle my shoe" );
* fun (); Answer: 1 3 4
  + uestion-
* 10 10 1O 10
* 5 10 10 5
* 10 5 5 10
* 5 10 10 10
* 5 5 5 5

Answer: 4

* + double total - find total (2.17, 3.0, 4.5. 3.145)
* Incorrect results will be produced because function Find Total () fails to initialize its variable
* The single parameter version of (IndTotal () is unnecessary, it will never be called.
* The recursion results in an infinite loop.
* Function findtotal () can take any number of parameters.
* There compile time errors wherever "\_." is used

Answer: 2 3 4 (2,3 don’t know; 4 correct)

* + In the C++ code excerpt below, which of the following are legal calls to the function template power ()?

template <typename T, int duration>

double power (double amplitude, int frequency=1000) (. . .)

* + - double output = power<int, 500> (3.0, 100) ;
    - double output = power<int, char> (3.0) ;
    - double output = power<int, 100> (3.0, "a') ;
    - double output - power<500> (3.0);
    - double output = power<int> (3.0, 100); Answer: 1 3
  + int foo(void) int 1;

for (i = -10; i < 10; +4i) process () ;

* 10
* 20
* 19
* 2
* 100

Answer: opt2-20

* + Which of the following statements accurately describe the C+ + code excerpt below? int\* arr new int[10];

delete arr;

* delete will fall
* The compiler will emit a diagnostic error.
* An exception may be thrown at runtime
* delete[] arr; must be used to properly deallocate arr.
* The free- store may be corrupted.

Answer: 4,5

Given the C++ class difinition and corresponding application code below, which of the following statements are true (the answer choices consider the presense or absense of Line D?

* If Line D is not commented out (exists), and properly implements move semantics this type with STL containers, the application might execute slower.
* Line Z will result in the constructor on Line C being called regardless of whether or not the constructor but (whether it exists or not).
* If Line D is commented out (does not exist) in the class definition result in
* If Line D is not commented out (exists), and properly implements that the object mas1 is intact the application co
* If Line D is commented out (does not exist) in the class definition as shown, then Line Y will result in being called. And, if Line D is not commented out (exists), then Line Y will result in the constructor o

Answer: 1 4 5

// 4,5

* + Which of the following correctly identify the base-10 value of x from the ANSI C Statements below?

int x=17; x>>=3; x<<=2;

* 10
* 8
* 11
* 2
* 16,

Answer: opt 2(val-8)

* + Question-
* p may point to memory outside the bounds of the array.
* The loop incorrectly increments the pointer when the correct element is located
* The const qualifier of p invalidates the pointer increment.
* The loop incorrectly increments the pointer before the array is access
* The loop incorrectly increments elements of the array through the pointer.

Answer: 3 4 5

* + Question-
* main's i and j are 100 200 change's i and j are 100 200 main's i and j are 100 200
* change's I and jare 10 20 main's I and j are 100 200 main's ) and | are 10 20
* main's I and ) are 10 20 change's I and | are 100 200 main's I and jare 10 20
* main's i and ] are 10 20 change's I and j are 10 20 main's i and | are 10 20
* change's I and ] are 100 200 main's I and ] are 10 20

in's ) and | are 10 20

Answer: 1 3 4

* + for(n = 0; n < 100; ++ n) a[n]=n=offset;

xmin = min; xmax = max;:

* The variables n, d and b lack the extern specifier, making them inaccess
* The variable d is not guaranteed to be nonzero.
* The static n is masked by an auto variable.
* The value of d is undefined
* init () fails to initialize the static n

Answer: 3

* + while (true)

int i= myFunc (j);

assert (i); // other omitted

* if the program is not compiled in debug mode, it does nothing.
* If i is a negative
* It logs the value of I to a file.65
* If the program compiled in debug mode, it causes program to terminate if the value of I is 0
* In debug mode it makes sure that I is within user-defined bounds and terminates the program it not.

Answer: 1 3 4

-

* + Which of the following ANSI C statements are valid definitions for a variable with a known initialized value?
* register short inc x3 = 3;
* auto int radius;
* register unsigned char c;
* static unsigned int bitmask,
* static char name [20]; Answer: 4

// 1,4.5

* + int main (int arge, char \*\*argv)

{

move (true) ; // Location #2

}

* Change the line at location #1 to: controllers: :spin (false) ;
* Change the line at location #2 to: controller: : horizontal : : move (true
* Change the line at location #1 to: horizontal: :spin (false);
* Change the line at location #2 to: horizontal; :move (true) ;
* Change the line at location #1 to: rotary: : spin (false) ; Answer: 5

-

* + Which of the following statements correctly describe usage of an unrestricted union in C+ + ?
* It must define a constructor.
* It has the unrestricted keyword qualifying the union name.
* It must define an assignment operator.
* It is allowed to contain members of class type.
* It is not allowed to contain members of built-in type.

Answer: 1 4

* + A usar opens an existing file for editing by issuing the 'vi my file' command and moves the cursor to the middle of the The editor is not in INSERT mode. Which of the following statements correctly describe the results of executing a +38/[01]yp/los/g command?
  + It will change all occurrences of "YP or yp' in the current line and the next 3 lines to "los".
  + It will change all occurrences of Yp' or 'yp' in the current line and the next 3 lines to "los".
  + It will change all occurrences uyp' or 'lyp' in the current line and the next 2 lines to los.
  + It will change all occurrences of 'uyp' or 'lyp' in the current line and the next 3 lines to los
  + It will change all occurrences of YP' or 'yp' in the current line and the next 2 lines to los
  + Table lookup and interpolation to approximate a mathematical function.
  + Operations on a large collection of database records, such as changing the order in which the ros
  + Calculation of cyclic redundancy checks.
  + Management of a list of character strings which reside in dynamically allocated memory.
  + Access of command-line options by main ().
  + 123... 255
  + 123…..127 -128 -127 -126…. -2 -1 0 1 2 … 127 -128 -127…. endlessly
  + 2 ... 255 0 1 2 3 254 255 ... endlessly
  + 123 127 128 ... 127 128... endlessly



1. Which of the following statements correctly describe the file command?

* It displays a complete summary of system information maintained about a file f
* It cannot be used with the name of a directory as an argument
* It allows the user to find a file in either the current directory or any of the die
* It uses the /etc/magic file to determine the meaning of the magic numbe
* It can display information about the contents of a file such as whether it is

answr: 4,5

* A for loop in which the loop index is declared in the initialization.
* A parameter list of an external function declared in a header file.
* An unsigned short defined inside a function combined with other quantities as a biosk
* A char array declared at file scope to be used by all functions defined in the file.
* Ashort int inside a block which will be passed by pointer to another function.

Answr: 1,4

//

1

* for (int room3: flooring ) displayFlooring (room3);
* for each (flooring.begin(), flooring.end()

[ ] (std::string room5) (displayFlooring (room5)|;

* for ( auto rooml: flooring)

displayFlooring (rooml) ;

* for std::string room2: flooring)

displayFlooring(room2);

* for (aute flooring: roomt) displayFlooring (room);

Answer: - 2,3,4

1. float beginningsalary new float() / float\* endingsalary - new floaty int feet new int[10](); int inches new int (101;

historystudent studenti - new historystudents historystudent student2 - new hlotorystudent 04

* beginningsalary and endingsalary are both initialized to zero.
* beginningsalary is zero, and endingsalary is unitialized.
* feet and inches are both arrays of 10 zeroes.
* feet is an array of 10 elements initialized to zeros and inches, is an arra
* The members of the object pointed to by student1 are uninitialized

//1,3

1. the following corectly define the function () according to the ANSI Coroar

* void Eoc(int p) (41)
* void foo(int "p[3] [41)
* void foo(int 'p[4])
* void foot (int \*p[] [4])
* void foo (int pf1 141)

// 5

1. Which of the following statements are correct after the highlighted command is executed? d-100 thr) aid-10 (progl

prog

100 Jun 27 1956 prog Sfile prog

Prog: 90xfamily dusand pagad pure escutable not stripped Sehnod, 6111 prog

* Any user executing pong will have al system privileges and restrictions of usering
* Only users who are members of the lang! group can execute meng
* Only users who are members of the rest group can change the patie \* prog file\*flie
* can ray bo ewecuted by root (super user)

// 1

* only user”Mr” will be able to exsone = /prog

1. Which of the Inding statorments about the ANSI C og torage som corect

* It enrols the declared object in a global list of variables for access by other transon unt
* Ita processor register cannot be allocated for the object at runtime, a compa
* It suggests that access to the object declared using the storage class be as f

// 3,5

* Objects declared with this storage class are initialized to zero before runtime
* it specifies one or more processor registers to be used to accommodate the oured object
* The code is erroneous: the loop index is not defined.
* The code is erroneous; the loop index ranges from 1 (anc) to size
* The code is erroneous; array elements are passed to art instead of a
* The code is correct and runs successfully,
* The code is erroneous; the values of the array are scanned through a loop Answr: 3

// 1

1. Which of the following property define an ANSI C function foo () which accepts as parameters an integer, an unsigned integer and a per to a character, and returns an unsigned integer?

* unsigned int too tint a, ansigned int b, char \*0)
* (unsigned int) Foo, int a, unsigned int 7, chat \* 1
* (int a. unsigned int y, char \*m) foo (unsigned int k\_
* Unsigned list foo (char \*q unsigned int x, int i)
* Foo(unsigned int r, int h, unsigned int j, char \*q)

1. Which of the following lieywords are required in order to utilize polymorphism in C++?

* pure
* polymorphic
* auto
* override
* virtual

1. SomeClass::data = 0; SomeClass obj Some;

* objSome::data
* SomeClass::data
* obj Some->data
* SomeClass::objSome.data
* SomeClass<static int>.data

1. When overloading C++unary operators, which of the following are legal options with respect to the number of parameters to be used?
   * Any number of parameters, when the operator function is a free standing tuhction (not a class member).
   * One (dummy) parameter, when the operator is a particular type of increment/decrement class member function.
   * No parameters, when the operator function is a class member
   * No parameters, when the operator function is a free standing function (not a class member)
   * One parameter, when the operator function is a free standing function (not a class member)
2. Which of the following are valid statements regarding the bitwise operator>>(right-shift) in ANSI C?
   * It prints two options: a true value and a random, false value.
   * It can be used with negative integers.
   * It can be used to reverse the sign of a number.
   * It has the effect of dividing an integer by powers of two.
   * It is a unary operator.
3. Which of the following ANSI C functions correctly set the least significant a bits of x to 1 (one), leaving all other bits uncha
   * Void f(void) { int i; for (i = 0; i < n; ++i) x << i; }
   * Void f(void){ x = 1 << n; }
   * void f(void) { int i; for (i = 0; i<n; ++i) x |= 1 << i; }
   * void f(void) {x |=~0<<n;}
   * Void f(void) (x1=-(-0 11x n); }
4. Box(double v) : volume (v) {}; Bool operator== (Box &rhs)

{

Return this->volume==rhs.volume;

}

Bool operator== (double v)

{

Box rhs (v);

return volume== rhs.volume;

}

};

* The existing overloaded operators already handle the third comparison.
* Implement a global function called operator which takes a double and a Box and have it call an existing men operator implementation with conforming parameters.
* Modify one of the existing member operator implementations so that it takes an additional parameter.
* Implement a global function called operator which takes a double and a Box and makes it a friend of c
* Implement a member operator taking a Box and a double.

1. Which of the following declaration statements must be added before the call to in the ANSI C program complies without error?

#include<stdio> Int main(void)

{

Int a

a= foot(10, 3.14) ;

printf(“%d”,a);

return 0;

}

foo(int aa, float bb)

{

return (float) aa + bb}

* Float footint, int);
  + int foo (int, float);
  + Float foo(float, int);
  + Int foo(float, float);
  + Float foo(float, float)

1. Which of the following statements about the ANSI C bitwise OR operator (1) are vald?
   * It can be used to set multiple tits of an integer.
   * It can be used to multiply an integer by powers of 2 (w)
   * It can be used to set a bit of an integer
   * Either coresponding input bit is 1 (one), the corresponding output bit is zero
   * It can be used to divide an integer by powers of 2 (two)

Answr: 1,3

1. If a base class in C++ overloads the assignment operator, which of the following steps need to be taken so that the operator is implemented in the derived classes?

Declare the overloaded operator in the base class to be

* + virtual.
  + Implement the overloaded operator as a friend to the base class.
  + Declare the overloaded operator in the base class to be dynamic.
  + None An operator overloaded in a base class is sufficient for all derived classes.
  + Override the operator in the derived class to account for any changes to the data members of the class.

1. Which of the following statements correctly describe the usage of operator overloading in C++?

* If a class overloads any operators, then all operators of the class must also be overloaded.
* An operator may not be overloaded more than once for a given class.
* It is possible to overload operators for integers.
* An overloaded operator may be implemented as a member function.
* Any operator may be overloaded.

1. When overloading C++ unary operators, which of the following are legal options with respect to the number of parameters to be used?

* Any number of parameters, when the operator function is a free standing function (not a class member).
* One (dummy) parameter, when the operator is a particular type of incrementidecrement class member function.
* No parameters, when the operator function is a class member.
* No parameters, when the operator function is a free standing function (not a class member).
* One parameter, when the operator function is a free standing function (not a class member).

1. Which of the following scenarios must a C++ destructor be declared virtual in a base class?

* When the default constructor of the base class is declared as virtual.
* When an implementation for the derived class destructor is not desired.
* When the developer wants to ensure that a derived class destructor is invoked when an instanc class pointer.
* When a derived class allocates system resources which are released in its destructor.
* When the destructor of the base class will be doing the clean-up of the derived class data. Page 31.

Printf(“\n%d”,c);

Return 0;

}

Void slaogan (void)

{

printf(“\n Only the best use c!”);

}

1. Error:’=’: cannot convert form ‘void ‘ to ‘int’.
2. Error: Undeclared identifier c.
3. Error: variable c must be of void type.
4. Error: void value cannot be assigned to identifier c.
5. Error: Function slogan() must return a value.

Output: 4

48. Which of the following statements regarding c++ exceptions are correct?

1. Exception handling can be safely delayed until the end of normal program execution.
2. Once handled, exceptions cannot be passed to next level for handling.
3. Exceptions can be handled by a catch() block.
4. Exceptions that are thrown, but not caught, can be ignored but might result in a logic error.
5. Exception can be thrown by a throw clause.

Output:345 Page 32.

Which of the following are NOT elements of exception processing in c++?

* 1. Making an exception specification using throw()
  2. Throwing an object of class Expection
  3. Using assert()
  4. Using dynamic\_cast<> on references
  5. Making an exception specification using noexcept

Output: 35

Which of the following are correct statements about C++ exception handling?

1. Exceptions not matching other catch blocks may be caught and handled by a handler specified

by catch(…)

1. An exception is routed to the most appropriate/matching catch block regardless of the order the catch blocks appear.
2. The most general catch block exception should appear last, with increasingly more specific catch

block exception pre….. it.

1. An exception handler with no exception parameter –such as catch() – will catch exception regardless if matching blocks also exist.
2. Multiple classes can be caught in a single catch clause as multiple arguments.

Output: 123

Which of the following actions are performed by the c++ statement below?

Catch(…)

1. Disable the trowing of further exceptions
2. Catch deault exxceptions
3. Ignore exceptions
4. Catch exception types that do not have a corresponding catch block.
5. Catch an exception, then pass it to the next level of program control.

Output:24

Throwing and catching C++ exception is an alternative to which of the following techniques?

1. Returning error codes.
2. Using hardware exceptions
3. Using assertions
4. Exhaustive debugging
5. Running the program with stack tracing enabled

Output:13

1. Which of the following statement about in ANSI c are correct?
   1. It is necessary to initialize the array at the time of definition
   2. The expression num[27] accesses the 28th element in the array
   3. The array int num[26] can store 26 elements
   4. The expression num[1] accesses the first element in the array
   5. The declaration num[SIZE] is allowed if SIZE evaluates to a constant

Output:3,5 Page 35:

1. P may point to memory outside the bounds of the array.
2. The loop incorrectly increments the pointer when the correct element is located
3. The const qualifier of p invalidates the pointer increment
4. The loop incorrectly incrementa the pointer before the array is accessed for the comp….
5. The loop incorrectly increments elementsof the array through the pointer.

Output: 345

Which of the following statements are correct about an array?

1. The array int num[26]; can store 26 elemets.
2. The expression num[1] designates the very first element in the array.
3. It is necessary to initialize the array at the time of declaration.
4. The declaration num[size] is allowed is the size is a macro.
   1. 1

b. 1,4

c. 2,3

d. 2,4

Answer: option b

Unsinged initialize(short \*p, short n )

{

Unsigned index = 0; While (index < n) P[index] = ++index; Return index;

}

1. The return value is off by one.
2. The array subscript in the loop is uncertain.
3. The array assignment syntax used on the pointer is invalid.
4. The type mismatch between n and index makes loop termination underfine.
5. The first value assigned by the loop will be “1” , not “0”

Output: 245

//2,5

The ANSI C functions below cause the program for when they are written to malfunction. Which of the following status describe defects in the code?

Static int n, d, b;

Static int xmin, xmax;

Static int a[100];

/\* Return the Y position of the point on the Line with a given x position within the x range: \*/ Int Y\_on\_line(int x)

{

if (x < xmin)

x = xmin; if(x > xmax)

x = xmax; return x\*n / d +b;

}

Void int(float scale, int offset, int min, int max)

{

Int n;

b = offset; n = scale;

if(scale < 1.0)

{

n\*= 100;

d = 100;

Class1 vUnit;

//other details omitted

}

# Options:

1. It is a short-hand notation for inheritance.
2. It allows for polymorphism.
3. It uses less memory that inheritance.
4. It gives myClass2 faster access to myClass1 than could be achieved via inheritance
5. It models a has-a relationship in myClass2 to myClass1.

# Default Answers: 145

**-------------------------------------------------------------------------**

Which of the following statements correctly describe a switch statement in an ANSI C program?

# Options:

1. A common set of statement cannot be used for multiple cases in a switch.
2. It is necessary to use a break in every switch statement.
3. A switch is used when one wishes to check the value of a variable against a particular set of values.
4. The controlling expression of the switch statement may be of any type.
5. A switch statement can offer be used instead of an if-else statement.

# Default Answers:2,3,5

**----------------------------------------------------------------------**

On a system in which an unsigned int occupies 4 bytes of memory, which of the following correctly represent changes in the value of the ANSI C pointer below caused by three consecutive increments of the pointer?

Unsigned int const \*p;

# Options:

1. The pointer value decreases by three.
2. The pointer value decreases by 12.
3. The pointer value increases by 4.
4. The pointer value increases by 12.
5. The pointer value increases by three.

# Default Answers:4

**-------------------------------------------------------------------------------------**

Which of the following represent definitions of an ANSI C pointer to char for which the dereferenced value may change, but the pointer value may not?

# Options:

1. Char const \* const p;
2. Const char \* const p;
3. Const char \*p;
4. Char \* const p;
5. Char const \*p;

# Default Answers: 4

In which of the following ANSI C programming scenarios may an array of pointers be used?

# Options:

1. Table lookup and interruption to approximate a mathematical function.
2. Access of command-line options by main().
3. Calculation of cycle redundancy checks.
4. Operations on a large collections of databases records, such as changing the order in which the records are accessed.
5. Management of a list of character strings which reside in dynamically allocated memory.

# Default Answers: 234

//2,5

**---------------------------------------------------------------------------------------**

# Options:

1) 6 5 9

2) 6 5 10

3) 5 5 9

4) 6 5 6

5) 5 4 5

# Default Answers: 4(6 5 6)

**-------------------------------------------------------------------------------------------------**

Char\* str = new char[20];

Which of the following statements can be used to free the memory?

# Options:

1. Delete[] str;
2. Delete str[];
3. Delete char\* str;
4. Delete \*str;
5. Delete str[20];

# Answers:1

**-------------------------------------------------------------------------------------**

Char \*p = “Hello”;

Char \* const q = “Hello”;

Const char \*r = “Hello”; Char const \*s = “Hello”; **Options:**

1. r is non-constant pointer to a string literal.
2. q can be made to point to another string literal.
3. s is a constant pointer to a non-constant string literal.
4. q is a constant pointer to a string literal.
5. p is defined as a non-constant pointer to a string literal.

# Answers:145

**---------------------------------------------------------------**

1. if(rtti (\*cp)).name() .== ”Taxi”) { startMeter(); }
2. if(typeid(\*cp) == typeid(c2)) { startMent(); }
3. if(typeid(cp) == typeid(c2)) { startMeter(); }
4. if(typeid(\*cp) == typeid(Taxi)) { startMeter(); }
5. if (typeid(c2) == typeid(Taxi)) { startMeter(); }

# Answers : 2 3 5

**--------------------------------------------------**

Which of the following statements are correct regarding std::weak\_ptr in C++?

1. Each instance does not increase the reference count of the pointer object being shared.
2. The object being referenced by the std::weak\_ptr instance must be checked to see if it still exists before it can be accessed.
3. Access to an instance’s contained pointer is via operator-> ().
4. An instance can transfer ownership of its contained pointer if assigned to an object of std::unique\_ptr.
5. An instance must be either an object of std::shared\_ptr or another std:: weak\_ptr object.

# Answers: 1 5

**---------------------------------------------------------------------**

36 page

1. Class singular : public singularity { Public:

Void getAnswer() = delete;

* 1. An effect of Line 4 is to prevent this particular overload of function getAnswer[int] from long implemented in the class.
  2. An effect of Line 5 is to remove the base class functionality of getAnswer() from the derived class.
  3. An effect of Lines 1 and 2 together is to make objects of this class not copy able.
  4. An effect of Lines 4 and 5 together is to ensure that all derived classes will only use the implementation on Line.
  5. An effect of Line 5 is to prevent any overloads of function getAnswer() from being implemented in the derived class.

Ans: 2,3,4

1. Which of the following C++ statements will result in a properly constructed object of class SomeClass or the free-store?
   1. new SomeClass\*;
   2. new SomeClass;
   3. calloc(size of(SomeClass),1);
   4. malloc(sizeof()SomeClass));
   5. realloc(0,sizeof(SomeClass),1);

Ans: 2, 3 and 4

// 2

1. Which of the following are valid expressions using the new operator in C++?
   1. Doctor &dref = new Doctor;
   2. Manager \*mptr = new Manager;
   3. Nurse nurses = new Nurse[5];
   4. int \*iptr = new int;
   5. Employee \*eptr = new Employee[10];

Ans: 2, 4 and 5

37 Page(Exception Handling)

1. The C++ unexpected () function is called in which of the following circumstances?
   1. When there is no appropriate catch block in the calling function, to catch an exception thrown by the called function.
   2. When something unexpected happens during normal program execution.
   3. When a function throws an exception which is not in the function definition’s throw list.
   4. When a function is called with parameters which do not match its declaration.
   5. When a function throws an exception unexpectedly, before a catch block is encountered.

Ans: 1, 2 and 3 38 Page

* + 1. Printf(“\n%d, c)

return 0;

}

Void slogan(void)

{

Printf(“\nOnly the best use C!”)

}

* + - 1. Error: ‘=’ cannot convert from ‘void’ to ‘int’
      2. Error: Undeclared identifier c
      3. Error: Variable c must be of void type.
      4. Error: void value cannot be assigned to identifier c
      5. Error: Function slogan() must return a value.

Ans: 4

* + 1. Which of the following statements regarding C++ exceptions are correct?
       1. Exception handling can be safely delayed until the end of normal program execution.
       2. Once handled, exceptions cannot be passes to the next level for handling.
       3. Exceptions can be handled by a catch () block.
       4. Exceptions that are thrown, but not caught, can be ignored but might result in a logic error.
       5. Exceptions can be thrown by a throw clause.

Ans: 3, 4 and 5

1. Page
2. Which of the following are NOT elements of exception processing in C++?
   1. Making an exception specification using throw ()
   2. Throwing an object of class Exception
   3. Using assert ()
   4. Using dynamic\_cast<> on references.
   5. Making an exception specification using noexcept.

Ans: 3 and 5

//5

1. Which of the following are correct statements about C++ exception handling?
   1. Exceptions not matching other catch blocks may be caught and handled by a handler specified

by catch(…)

* 1. An exception is routed to the most appropriate/matching catch block regardless of the order the catch blocks appear.
  2. The most general catch block exception should appear last, with increasingly more specific catch block exceptions.
  3. An exception handler with no exception parameter- such as catch () – will catch any exception regardless if matching.
  4. Multiple classes can be caught in a single catch clause as multiple arguments.

Ans: 1, 2 and 3

// 1,3

1. Which of the following actions are performed by the C++ statement below?

Catch(…)

* 1. Disable the throwing of further exceptions
  2. Catch default exceptions
  3. Ignore exceptions
  4. Catch exception types that do not have a corresponding catch block
  5. Catch an exception, then pass it to the next level of program control.

Ans: 2 and 4

// 2,4,5

1. Page
2. Throwing and catching C++ exceptions is an alternative to which of the following techniques?
   1. Returning error codes
   2. Using hardware exceptions
   3. Using assertions
   4. Exhaustive debugging
   5. Running the program with stack tracing enabled.

Ans: 1 and 3

ARRAYS

1. Which of the following statements about arrays in ANSI C are correct?
   1. It is necessary to initialize the array at the time of definition.
   2. The expression num[27] accesses the 28th element in the array
   3. The array int num[26] can store 26 elements.
   4. The expression num[1] accesses the first element in the array
   5. The declaration num[SIZE] is allowed if SIZE evaluates to a constant.

Ans: 3 and 5

2,3,5

* + 1. Which of the following are types of polymorphism in C++?
* Parametric polymorphism
* Hybrid polymorphism
* Bi-metric polymorphism
* Single polymorphism
* Post processing polymorphism

1. AC++ developer wants to create a derived class which inherits base class members such that its private inte in the derived class and its protected and public members have protected scope in the derived class. Which of the stag declarations can the developer use to achieve the objective?

* Virtual public inheritance
* Protected inheritance
* Private inheritance
* Virtual private inheritance
* Public inheritance

1. Which of the following are true statements about C++ inheritance access specifications?

* Public inheritance from the base class makes all the base class members public in the deved class
* Private members in the base class are not changed via the inheritance access specifications.
* Protected inheritance from the base class makes all the base class members protected in the derived trans
* Private inheritance from the base class makes all the base class members private in the derived class
* Dynamic inheritance from the base class allows derived class member's access control to be changed on the

1. Which of the following statements regarding C++ exceptions are correct?

* Exception handling can be safely delayed until the end of normal program execution.
* Once handled, exceptions cannot be passed to the next level for handling.
* Exceptions can be handled by a catch() block.
* Exceptions that are thrown, but not caught, can be ignored but might result in a logic error.
* Exceptions can be thrown by a throw clause.

1. void fune () ()

);

//…

void example () Derived objDerived; //statement goes here

* Base: : func();
* Base: : Derived: : func();
* obj Derived.Base: : func();
* Derived: : func();
* obj Derived: : func();

1. return U; void slogan (void)

};

void ex

printf("\n Only the best use C!");

* Error: Function slogan() must return a value.
* Error: void value cannot be assigned to identifier c.
* Error: Undeclared identifier c.
* Error: '': cannot convert from 'void' to 'int'.
* Error: Variable c must be of vold type.

1. of the application has been omitted

* If Line D is commented out (does not exist) in the class definition as shown, then Line Y will result in a compiler error.
* If Line D is not commented out (exists), and properly implements move semantics, and an application stores and uses objects of this type with STL containers, the application might execute slower.
* Line Z will result in the constructor on Line C being called regardless of whether or not the constructor on Line D is commented out (whether it exists or not).
* If Line D is not commented out (exists), and properly implements move semantics, the application code after Line Y can assume and that the object mas1 is intact.
* If Line D is commented out (does not exist) in the class definition as shown, then Line Y will

result in the constructor on Line C being called. And, if Line D is not commented out (exists),

then Line Y will sult in the constructor on Line D being called.

* Modify one of the existing member operator Implementations so that it takes an additional parameter
* The existing overloaded operators already handle the third comparison.
* Implement a member operator taking a Box and a double.
* Implement a global function called operator which takes a double and a Box and makes it a friend of class
* Implement a global function called operator which takes a double and a sox and have it call an existing member operator implementation with conforming parameters

1. WARMER A value, multiplied by m. Otherwise, return -1: \*/ int lookup (int a, int 1, int v, int m)

int i = 0;

while (i++ < 1) if (a + i = v) return v m; return -1;

* The while loop encloses the if statement, but not the return.
* There is no path of execution that does not pass through return -1;.
* The array index i is improperty incremented between the loop test and the array comparison
* is of the wrong type for pointer arithmetic, it must be unsigned.
* The intended pointer arithmetic is not performed.

1. class Readquarters public Department, public Division (1: main ()

HeadQuarters hq;

* Declare class Headquarters such that the inheritance order is reversed, putting Division first, (e.,

publie sivis public Department) then refer to member building via objent hq in main() without

further qualification such hq.building = 'A';

* In main (), when using object hq, refer to member building via the Division base class as a member, and member of the subclass such as: hq.Division.building = 'A':
* In class Headquarters, refer to member building by including the name of the desired base class such as Division::building = 'A';
* In class Headquarters, add the qualifier "using Division: building;" with public access level then refer to member building from main() via the object name and member, without further qualification, such as: hq.building

1.

class wingulator : publie singularity ( publin:

void getAnswer() delete;

// tine 5

* + An effect of Line 5 is to remove the base class functionality of getanover from the deved class
  + An effect of Lines 4 and 5 together is to ensure that all derived classes will only use the implementation
  + An effect of Line 5 is to prevent any overloads of function getAnrver() from being implemented
  + An effect of Lines 1 and 2 together is to make objects of this class not copyable
  + prevent An effect of Line 4 is to prevent this particular overload of function getaner(in hom being imple class

1. int main(int arge,

char \*\*argv)

move (true);

// location #2

* Change the line at location #1 to: controllers::spin (false);
* Change the line at location #2 to: controllers::horizontal::move (true
* Change the line at location #1 to: horizontal::spin (false);
* Change the line at location #2 to: horizontal::move (true);
* Change the line at location #1 to: rotary::spin (false);
* 1)Which of the following correctly define the ANSI C line below? char \*token(char\*, int, char);
  1. A function initializer
  2. A function prototype
  3. A return value
  4. A function declaration
  5. A type definition Answr : 2,4
  6. Which of the following ANSI C statements are valid definitions for a variable with a known initialized value?
     1. auto int radius;
     2. register short int x3 = 3;
     3. static char name[20];
     4. register unsigned char c;
     5. static unsigned int bitmask; Answr: 2,5
  7. Which of the following correctly describe a char variable in ANSI C?
     1. The format specifier %c is used with printf() to output a char variable as a character.
     2. A char variable can store either an ASCII character or a Unicode character.
     3. The modulus operator(%) cannot be used with a char variable.
     4. A char variable is large enough to store any number.
     5. Both a carriage return(\r) and a value are stored if scanf() used to store the variable in a char variable

Ans:-1,2,3

1. Which of the following are differences between signed and unsigned variations of the same

integer type in a 2’s complement representation of ANSI C?

* 1. The range of the signed and unsigned variations of the same integer type are equivalent in size and numeric maximum and minimum.
  2. The range of the unsigned variation of an integer type is twice that of a signed variation.
  3. A negative value, expressed as a signed integer, will be expressed as positive value greater than the maximum value of the signed integer, if expressed as an unsigned integer of the same size.
  4. They have the same size, but the signed variation range is centered around 0. And athe unsigned variation range has a lower bound of 0.
  5. The representation of a signed integer has one additional bit for the sign. Ans:-3

1. Assuming a 2’s complement implementation of ANSI C, which of the following lines of code will generate an output of -32768?
   1. char m = 32768; printf(“%d”, m)
   2. unsigned long int l = 32768; printf(“%d”, l);
   3. unsigned short int k = 32768; printf(“%d”, k);
   4. short int i = 32768; printf(“%d”, i);
   5. long int j = 32768; printf(“%d”, j );

Ans:-4

1. Which of the following are valid ANSI C statements? a)int i = 35; i = i %5;
2. long double k = 1.25; k = k %2;
3. double d = 6.28; d = d%4;
4. long float a = 3.14; a\*=2;
5. short int j = 255; j = j%15; Ans:-1,4,5
6. Which of the following correctly describe the usage of a long double data type in ANSI C? a)a long double is a type of integer
7. A long double can be printed using the format specifier %Lf.
8. The modulus operator can be used with a long double.
9. A long double occupies 4 bytes of memory.
10. A long double should be used if the range of a double does not accommodate the quantities to be expressed.

Ans:-2,5

1. int j = 5; while(true)

{

int I = myFunc(j); assert(i);

//other code omitted

}

* 1. If the program is not compiled in debug mode, it does nothing.
  2. If I is a negative value it terminates the program.
  3. It logs the value of I to a file.
  4. If the program is compiled in debug mode, it causes the program to terminate if the value of I is 0.
  5. In debug mode it makes sure that I is within user-defined bounds and terminates the program if not. When not in debug mode logs the value of I to a file.

Ans:-1,3,4

1. In which of the following scenarios must a c++ destructor be declared virtual in a base class? a)When the default constructor of the base class is declared as virtual.
2. When an implementation for the derived class is destructor is not desired.
3. When the developer wants to ensure that a derived class destructor is invoked when an instance class pointer.
4. When a derived class allocates system resources which are released in its destructor.
5. When the destructor of the base class will be doing the clean-up of the derived class data. Ans:- 1,5
6. Which of the following ANSI C statements are valid definitions for a variable with a known initialized value?
7. register unsigned char c;
8. register short int x3 = 3;
9. static unsigned int bitmask ;
10. static int radius;

Ans:-3

1. return 0;

}

void slogan(void)

{

printf(“\n Only the best use C!”);

}

1. Error: Function slogan() must return a value .
2. Error: Void value cannot be assigned assigned to identifier c.
3. Error: Undeclared identifier c.
4. Error: ‘=’ cannot convert from ‘void’ to ‘int’.
5. Error: Variable c must be of void type. Ans:-2.

q) other details omitted

1. It is a short-hand notation for inheritance.
2. It allows for polymorphism,
3. It uses less memory that inheritance.

iV) It gives myClass2 faster access to myClass1 than could be achieved via inheritance

1. It models a has-a relationship in myClass2 to myClass1.

Answr : 1 4 5.

Q) Which of the following are accurate statements concerning the behavior of the dynamic\_cast operator in C++?

"i) The dynamic\_cast operator's behavior is a subset of the reinterpret\_cast operator's behavior.

* 1. The dynamic\_cast operate can be applied to a pointer or reference to a polymorphic object
  2. A dynamic castapplied to reference throes an appexception if the operation fails.
  3. The dynamic cast operator fails if used forupcasting.
  4. A dynamic\_cast applied to a pointer returns 0 (zero) if the operation fails.

Answr : 1,2,5

Q) Which of the following are types of polymorphism in C+?

1. Parametric polymorphism
2. Hybrid polymorphism
3. Bi-metric polymorphism
4. Single polymorphism
5. Post processing polymorphism

Answr: 1

Q) A C+ + developer wants to create a derived class which inherits base class members such that its private

in the derived class and its protected and public members have protected scope in the de declarations can the developer use to achieve the objective?

1. Virtual public inheritance
2. Protected inheritance
3. Private inheritance
4. Virtual private inheritance
5. Public inheritance Answr: 2

Q) Which of the following are true statements about C++ inheritance access specifications?

1. Public inheritance from the base class makes all the base class members public in the derived class,
2. Private members in the base class are not changed via the inheritance access specifications.
3. Protected inheritance from the base class makes all the base class members protected in the derived class.
4. Private inheritance from the base class makes all the base class members private in the derived class.
5. Dynamic inheritance from the base class allows derived class member's access control to be changed on the fry

Answr:-2,3,4

Q) Which of the following statements regarding C++ exceptions are correct?

1. Exception handling can be safely delayed until the end of normal program execution.
2. Once handled, exceptions cannot be passed to the next level for handling.
3. Exceptions can be handled by a catch () block.
4. Exceptions that are thrown, but not caught, can be ignored but might result in a logic error.
5. Exceptions can be thrown by a throw clause.

Answr: 3,4,5

public:

void func () ()

1 1 . . .

void example () ( Derived objDerived;

/ /Statement goes here

i) Base : : func () ;

1. Base : : Derived: : func () ;
2. objDerived. Base: : func () ;
3. Derived: : func () ;
4. obj Derived: : func () ;

answr: 3

1. return U;

void slogan (void)

printf ("\n Only the best use c!") ;

* 1. Error: Function slogan( ) must return a value.
  2. Error: void value cannot be assigned to identifier c.
  3. Error: Undeclared identifier c.
  4. Error: '=: cannot convert from 'void' to 'int'.
  5. Error: Variable c must be of void type.

Answr:2

Q)

rest of the application has been omitted

1. If Line D is commented out (does not exist) in the class definition as shown, then Line Y wil result in a complex error.
2. If Line D is not commented out (exists), and properly implements move semantics, and an application stores and uses objects of

this type with STL containers, the application might execute slower.

1. Line Z will result in the constructor on Line C being called regardless of whether or not the constructor on Line D is commented

out (whether it exists or not).

1. If Line D is not commented out (exists), and properly implements move semantics, the application code after Line Y can assume

that the object masi is intact.

1. Line D is commented out (does not exist) in the class definition as shown, then Line Y will result in the constructor on Line C

being called. And, if Line D is not commented out (exists), then Line Y will result in the constructor on Line D being caled.

Answr: 2,4,5

1. return volume = rhe. volume;
   1. Modity one of the existing member operator = Implementations so that it takes an additional parameter.
   2. The existing overloaded operators already handle the third comparison.
   3. Implement a member operator= taking a Box and a double.
   4. Implement a global function called operator= which takes a double and a Box and makes it a friend of class BL.
   5. Implement a global function called operator = which takes a double and a Box and have it cal an existing member operator== implementation with conforming parameters.

Answr:4

Q) value, multiplied by m. Otherwise, return -1: ./ int lookup (int \*a, int 1, int v, int m)

int i = 0;

while (it+ < 1) if ('a + i = v) return v \* m; return -1;

1. The while loop encloses the if statement, but not the return.
2. There is no path of execution that does not pass through return -1;.
3. The array index i is improperly incremented between the loop test and the array comparison.
4. i is of the wrong type for pointer arithmetic; it must be unsigned.
5. The intended pointer arithmetic is not performed.

Answr: 3,5

q) class Headquarters : public Department, public Division ( 1; main()

HeadQuarters hq;

1. Declare class Headquarters such that the inheritance order is reversed, putting Division first, (Le, public Division,

public Department) then refer to member building via object ng in main () without further qualification such as:

hq. building = 'A';

1. In main (), when using object hq, refer to member building via the Division base class as a member, and building as a

member of the subclass such as: hq. Division. building = 'Al;

)In class Headquarters, refer to member building by including the name of the desired base class such as;

Division: :building = 'A";

1. In class Headquarters, add the qualifier "using Division: :building;" with public access level then refer to member
2. In class headquarters ad the qualifier using division building from main () via the object name and member, without further qualification, such as. hq. building = "Wi

Answr: 1,2,3

1. int main (int argc, char .\*argv) move (true) ;

// location #2

* 1. Change the line at location #1 to: controllers: : spin (false) ;
  2. Change the line at location #2 to: controllers: : horizontal: : move (tru iii) Change the line at location #1 to: horizontal: : spin(false) ;

1. Change the line at location #2 to; horizontal: :move (true) ;
2. Change the line at location #1 to: rotary: : spin (false) ;

Answr: 5

Which of the following statements correctly describes the results of executing the highlighted command below?

$id

Uid-108(mir) gid=10 (prog)

$id -1

-rw-rw-rw- 1 mr prog 188 Jun 27 1996 my\_file

$chmod 754 my\_file

$ls-l

|  |  |  |  |
| --- | --- | --- | --- |
| 1) -rwxrw-r-- | 1 mr | prog | 188 Jun 27 1996 my\_file |
| 2) -rwxr-xr-- | 1 mr | prog | 188 Jun 27 1996 my\_file |
| 3) –rw-rw-rw | 1 mr | prog | 188 Jun 27 1996 my\_file |
| 4) –rwxr-x--x | 1 mr | prog | 188 Jun 27 1996 my\_file |
| 5) w--wx | 1 mr | prog | 188 Jun 27 1996 my\_file |
| Answer: - 2 |  |  |  |

#include<stdio.h> Int foo(void)

{

Printf(“foo”);

Return 2;

}

Int bar(void)

{

Printf(“bar ”);

Return 3;

}

Int main(void)

{

Int a = 1;

Printf(“%d”, a? foo() : bar());

Return 0;

}

1. Bar 3
2. 3 foo bar
3. 2 foo bar
4. Foo bar 2
5. Foo 2

Answer : - 5

1. Begin slow end
2. Begin slow halt waiting
3. Begin slow halt waiting end
4. Begin slow halt
5. Begin end

Answer: - 3

Which of the following statements regarding the C++ 11 operators alignof() and alignas() are correct?

1. Alignas() works on expressions.
2. Alignas() returns the size, in bytes, of the expression in parentheses.
3. Alignas() is used to specify custom alignment of variables and user-defined types.
4. Alignof() is used to obtain the alignment of a specified variable or type.
5. Alignof() returns the size, in bytes, of the expression in parentheses. Answer: - 1,4,5

//4,5

1. R R R
2. R0 V0 X0
3. R0 R150 R50
4. R0 R0 R0
5. R0 V150 S50

Answ: 4,5

1. The output will be: e1 e1 e1
2. There will be no output.
3. The output will be: e2 e1 e2
4. There will be a runtime error due to missing E2 carch block.
5. The code will not compile due to missing E2 catch block. Answer: - 2,4

If a C++ program continues to allocate memory from the free store with new operator but never releases that memory with the delete operator, which of the following can result?

1. Increasing amounts of system memory will become unavailable to other processes until the system is rebooted.
2. Eventually an attempt to allocate more memory with the new operator will result in an exception.
3. Increasing amounts of system memory will become unavailable to other processes until the garbage collection routine.
4. When the maximum allowable memory has been allocated (on a per program basis) the system will halt the program.
5. Eventually an attempt to allocate more memory with the new operator will already allocated, but unused,

Answer: - 2,4,5

// 1,2,4

Which of the following are valid statements regarding the ANSI C bitwise AND operator (&)?

1. It has no effect on the sign bit of an unsigned integer.
2. It can be used to divide an integer by a power of 2(two).
3. If either corresponding input bit is 1 (one), the corresponding output bit is 1(one).
4. It can be used to determine the address of the operand.
5. It can be used to check whether a bit of an integer is set or clear. Answr: 3,4

Char const \*month (int m)

{

If(m<0) m = 0;

`if (m > sizeof months) m = sizeof months;

return months[m];

}

1. Month () incorrectly limits the value of m, allowing access beyond the end of the array.
2. The return type of the function is correct; it should be char\* const month(int);
3. The length of the array is undefined, therefore no upper bound on m can be really enforced.
4. The return type of the function is incorrect; it should be char const \*\*month(int); .
5. The array has no terminator, so its end cannot be found. Answ: 3

// 1,3,4

What of the following are valid statements regarding pointers in the ANSI C programming language?

1. A pointer is a variable that contains the address of another variable.
2. If a pointer is incremented to a variable of a different type, the type of the pointer automatically changes that of the new type.
3. To point to consecutive data objects in memory, a pointer is incremented by the size of the data object.
4. Pointer arguments allow a function to access a variable defined somewhere else in a program.
5. Pointers must be declared with global scope. Answer: - 1,3,4

Class Level15 : public Level13

{

Public:

Double func2(char) override; Virtual double func3(int) = delete; Virtual double func4(char) override;

};

1. Level11 : : func1 () functionality has been removed from Level2 but is still available to subsequent levels.
2. Level5 results in a compiler error since it cannot inherit from Level13.
3. Level15 : :func4(char) will produce a compiler error since it is not a true override.
4. Level15 : : func3(int) will produce a compiler error since virtual functions implemented elsewhere cannot be deleted.
5. Level15 : : func2(int) will override any functionality defined in Level12: : func2(int).

Which of the following statements correctly describe the ANSI C multiplication (\*) operator?

1. It produces the product of its two operands.
2. Its operands must either both be signed or both unsigned.
3. It binds more tightly than the addition (+) and subtraction (-) operators.
4. Its operands must be of the same of the same type and precision.
5. It binds more tightly than the % operator. Answer: - 1,3

Which of the following statements correctly describe the ANSI C program follow? #include<stdio.h>

Int = 10;

Int main(void)

{

Int x =20;

{

Int x=30;

Printf(“%d”,x);

}

Printf(“%d”,x);

Return 0;

}

1. Two local variables x cannot be defined inside main().
2. The output will be: 30 30
3. The program will report an error: “Redefinition of variable x”.
4. The output will be: 30 20
5. Variables inside nested blocks must be explicitly declared auto. Answer:- 4

Int main()

{

xxx\*bptr = new YYY; YYY \* dptr;

\*\*\*\*\*

}

1. Dptr = bptr;
2. Dptr = dynamic\_cast<XXX\*>(bptr);
3. Dptr = static\_cast<YYY\*>(bptr);
4. Dptr = up\_cast<XXX\*>(bptr);
5. Dptr = dynamic\_cast<YYY\*>(bptr); Answer: - 1,5

Which of the following are valid statements regarding a pointer in ANSI C?

1. Decrementing a pointer into an array causes it to refer array element with the next lower subscript.
2. Automatic boundary checking prevents pointers from pointing outside the array to which they refer.
3. The pointer must point to the data object or into array to which it was originally assigned.
4. Decrementing a pointer to a non-array variable causes the variable to be decremented instead of the pointer.
5. A pointer may point to a point to a data object located at any valid address in memory. Answer: - 1,5
6. It will not compile since structures are not allowed to have member functions.
7. There is no difference between the class and structure as defined.
8. The structure has no private data members.
9. The class version will perform much faster than the structure version.
10. The class has no private data members. Answer: - 2,5

Which of the following will be the output of executing the ANSI C program below?

#include<stdio.h> int main(void)

{

Int a[5] ={5, 1, 15, 20, 25};

Int \*p = &a[2];

Printf(“%d\n”p[-1]); Return 0;

}

1) 0

2) (An undefined value) 3) 1

4) 15

5) 5

Answr: 3

1. If an implementation for cleanup() is provided in class WaterSystem, it will be called by the WaterSystem destruct an object of type PumpUnit is destroyed.
2. If an implementation for cleanup() is provided in class WaterSystem, the implementation for cleanup() in Public be called by the WaterSystem destructor when an object of type PumpUnit is destroyed.
3. The implementation of cleanup() in the class PumpUnit cannot be called by the destructor in WaterSystem.
4. When an object of type PumpUnit is destroyed, the virtual function cleanup() in PumpUnitwill be called by the WaterSystem destructor.
5. If an implementation for cleanup() is provided in class WaterSystem, the implementations for cleanup() WaterSystem and PumpUnit will be called by the WaterSystem destructor when an object of type Pump[].

Which of the following statements correctly describe the value of the ANSI C expression below? x % y

* 1. It is 1(one) when x and y are equal.
  2. It is the remainder of the division of x by y.
  3. It is 0(zero) when y divides x exactly.
  4. It is y when y divides x exactly.
  5. It is 50 when x is half of y. Answer: - 2

Which of the following ANSI C keywords may cause the loop in which they appear to terminate prematurely?

1. Default \*\*//options will be same for the aother question is looping construct then the answr is (continue)\*\*
2. Break
3. Continue
4. Until
5. Goto

Answer:- 5,2

1. The storage class of the variable i is static.
2. The variable i is a global variable and is available to both main() and val().
3. The variable i used in main() is different from the one used in val().
4. The variable i must be passed to val() in order to become available to theval()
5. The variable I used in both main() and val() is stored in different memory location.

Which of the following statements correctly describe functions in ANSI C?

1. Every function must return a value.
2. Functions cannot return more than one value.
3. The maximum number of arguments that a function can take is 12.
4. The names of functions in two different files which files which are linked together do need to be unique.
5. Functions can be called by a value or pointer. Answer:- 2,5

1) X = 4, y = 5

2) X=20,y=18

3) X=8,y=10

4) X=10, y=10

5) X=16, y=10

Which of the following statements correctly describe attributes of overloaded functions in C++?

1. The type void is an overloaded type. For example:

function(void); //can also be written as function(); function(int);

1. Only the last of the functions’ arguments can be different.

For example:

function(int, float); function(int, char);

1. The functions’ arguments must have at least one type different.

For example:

Function(float, int); Function(char, int);

1. The functions’ arguments must have at least one type different. For example:

function(int); function(float);

1. A typedef can be used create deferent overlaodable types. For example:

typedef int A; typedef int B; function(A); function(B); Answer:- 1,4

which of the following statements correctly describe the ‘cu’ command?

1. In order to break the connection between two systems during a ‘cu’ operation, the user can

enter- . At the beginning of a new line.

1. While using it the user can execute an ‘ls’ command on the local system by entering -! ls at the beginning of a new line
2. It can only be executed by root (super user).
3. It avoids the needs to know a password on remote systems.
4. It can be used to transfer data between two systems by using the –p and –t commands. Answer: - 3,5

Q) Which of the following statement about the ANSI c register storage classes are correct?

1. it enrolls the declared object in aglobal list….
2. if a processer register cannot be allocated for the object.
3. it suggest that access to the object declaration using the storage. iv)object declared with this storage class are initiazed to zero.

v)it specifies one or more processor register to be used to accommodate.

Answr: 1

1. given the ansi c definitions below which of the following expression int a[5];

int \*pa;

* 1. Pa = \*a;
  2. Pa = a;

3) a++;

4) pa = &a;

5) pa = &a[0];

answr: 2,4,5

// 2,5

1. which of the following statements are true about c ++lamba functions?
   1. they can be only used in a loop to operate
   2. they can be given access
   3. they are similar to unnamed
   4. they cannot return a value
   5. they are essential to object –oriented programming

Answr: 2,3,5

1. which of the following statements about the use of the virtual functions
   1. a virtual function needs to be declared
   2. a static function can be declared
   3. an operater can be declared
   4. a data member can be declared
   5. a friend function can be declared

Answr: 1,3

1. what of the following statements accurately describe the ANSI c does-while loop?
   1. do-while loops may be nested only inside
   2. the loop may be used to construct an
   3. the until keyword may be used inplace
   4. multiple statements inside the body of the loop
   5. the loop will execute the statements in its body at least..

Answr: 2,5

1. in which of the following aceniors may pointers be utilized in ANSI c?
   1. A set of functions which collect data from sensors and adjust an output value
   2. A set of function implementing useful mathematical function
   3. A set of function which search through data records stored in memory as structs…
   4. A set of function responsible for handling a set of data
   5. A set of functions which compare elements of an array

Answr:3,4,5

,,q)

ar in which the ANSI C extern keyword

* + 1. A library of functions defined in a single translation unit, with variables defined once which must be accessible to the library.
    2. Aprogram arranged with functions in many translation units and all global data together in one translation unit.
    3. A library of functions implementing a mathematical relationship using a lookup table stored in read-only memory
    4. Alunction that calls another function in a different translation unit, for which no declaration exists for the called header file or otherwise.
    5. A library of functions defined in a single translation unit, with data objects which must be exposed to functions

Answer: - 2,5

le some program there is an object named nyt, which is an instantiation of a clase de cosa a operit comst talics on Liv and returns Which of the following are true statements about the implementatiohe pass the operatie sed tage of the operator?

* + - 1. The class can also implement another version of the overloaded operator () Such one which takes two
      2. The class may have implemented the overloaded operator) as either a mem function of non-member: function
      3. The class can also implement another version of the overloaded operotor (1h as one which takes an la and 14
      4. Implementations of the overloaded operatur may not access data memb of the class.
      5. Given the object named mych as described, this is a legal way to call its ovoaded operator 1 by

Answers: - 2,3

conectly describe heap.

* + - * 1. Local variables are stored on the stack.
        2. Memory from the heap can be requested by the new operator.
        3. Stack-based variables can be accessed faster than heap-based variables.
        4. Stack-based variables must be freed when they are no longer used whereas heap variable are no longer needed.
        5. The size (amount of memory required) for stack-based variables must be known at complet

Answer: - 1,2,5

,3

Which of the following statement about in ANSI c are correct?

1. It is necessary to initialize the array at the time of definition
2. The expression num[27] accesses the 28th element in the array
3. The array int num[26] can store 26 elements
4. The expression num[1] accesses the first element in the array
5. The declaration num[SIZE] is allowed if SIZE evaluates to a constant
   1. 1

b. 1,4

c. 2,3

d. 2,4

Answer:- b

Which of the following statements correctly describe the associativity of the operators used in A

1. In the expression ab 5, the order of assignment is not decided by the associativity of the operaties
2. Two different operators always have different associativity.
3. Every operator has an associativity
4. The associativity of an operator is either left-to-right or right-to-left.
5. Associativity does not play a role unless the precedence of the operators is the same. Answer: - 3,4,5

Q)

* 1. Change the type of x to a type larger than…
  2. Remove the extraneous various b…
  3. Change the return value
  4. Change if (x | c) to if ..
  5. Change ! (x & 0x100) to…..

Answr) 3,4,5

q)

1) the size of a shot int and a long int vary from oneplatform to …

2) a short int is 4 bytes wide,

3) both shot int and long int

4) the size of the int and a long int can be verified..

5) a short int is 2 bytes..

Answr) 1,4