Why .Net Does not Support multiple inheritance?

when we swipe the credit card what will genertaed in Back end. how system will validated from the back end for credit card who is Aquriree and Issuer ,

You just entered the following command_Router(config#) line console 0 Which operation is most likely to follow? A. Confound terminal type B. enter protocol parameters for a serial line C. create a password on the console terminal line D. establish a terminal type 4 connection to a remote host E. change from configuration mode to console privileged mode

int f() { int I = 12; int &r = I; r += r / 4; int *p = &r; *p += r; return I; } Referring to the sample code above, what is the return value of the function "f()"? a) 12 b) 15 c) 24 d) 17 e) 30

catch(exception &e) { . . . } Referring to the sample code above, which one of the following lines of code produces a written description of the type of exception that "e" refers to? a) cout << e.type(); b) cout << e.name(); c) cout << typeid(e).name(); d) cout << e.what(); e) cout << e;

class X { public: int x; static void f(int z); }; void X::f(int y) {x=y;} What is the error in the sample code above? a) The class X does not have any protected members. b) The static member function f() accesses the non-static z. c) The static member function f() accesses the non-static x. d) The member function f() must return a value. e) The class X does not have any private members.

The "virtual" specifier in a member function enables which one of the following? a) Monmorphism b) Late binding c) Metamorphism d) Solomorphism e) Inheritance

Which one of the following describes characteristics of "protected" inheritance? a) The base class has access only to the public or protected members of the derived

class. b) The derived class has non-public, inheritable, access to all but the private members of the base class. c) The derived class has access to all members of the base class. d) The private members of the base class are visible within the derived class. e) Public members of the derived class are privately accessible from the base class.

class Foo { public: Foo(int i) { } }; class Bar : virtual Foo { public: Bar() { } }; Bar b; Referring to the above code, when the object 'b' is defined, a compiler error will occur. What action fixes the compiler error? a) Adding a virtual destructor to the class Bar b) Adding a constructor to Bar which takes an int parameter c) Adding "Foo()" to the Bar constructor d) Adding a copy constructor to the class Foo e) Adding "Foo(0)" to the Bar::Bar initializer list

class Foo { const int x; protected: Foo(int f); ~Foo(); }; Foo f; Referring to the sample code above, why will the class declaration not compile? a) The variable x is const. b) The destructor is protected. c) The destructor is not public. d) The constructor is protected. e) There is no default constructor.

class X { private: int a; protected: X(){cout<<"X constructor was called"<<endl;} ~X(){cout<<"X destructor was called"<<endl} }; Referring to the code above, which one of the following statements regarding "X" is TRUE? a) X is an abstract class. b) Only subclasses of X may create X objects. c) Instances of X cannot be created. d) X objects can only be created using the default copy constructor. e) Only friends can create instances of X objects.

class HasStatic { static int I; }; Referring to the sample code above, what is the appropriate method of defining the member variable "I", and assigning it the value 10, outside of the class declaration? a) HasStatic I = 10; b) int static I = 10; c) static I(10); d) static I = 10; e) int HasStatic::I = 10;

class Foo { int x; public: Foo(int I); }; If a class does not have a copy constructor explicitly defined one will be implicitly defined for it. Referring to the sample code above, which one of the following declarations is the implicitly created copy

constructor? a) Foo(Foo *f); b) Foo(Foo &f); c) Foo(const Foo *f); d) Foo(const Foo &f); e) Foo(int);

class Alpha { public: char data[10000]; Alpha(); ~Alpha(); }; class Beta { public: Beta() { n = 0; } void FillData(Alpha a); private: int n; }; How do you make the above sample code more efficient? a) If possible, make the constructor for Beta private to reduce the overhead of public constructors. b) Change the return type in FillData to int to negate the implicit return conversion from "int" to "void". c) Make the destructor for Alpha virtual. d) Make the constructor for Alpha virtual. e) Pass a const reference to Alpha in FillData

How long does this loop run: for(int x=0; x=3; x++) a) Never b) Three times c) Forever

1. What does the following do: void afunction(int *x) { x=new int; *x=12; } int main() { int v=10; afunction(&v); cout<<v; } a) Outputs 12 b) Outputs 10 c) Outputs the address of v

What is the Difference between Volume Testing and Load,Stress tests..?

26) The device that can transform digital data into analog data is called a

a)transformer

b)network

c)carrier

d)modem

Ans. d

27) Following is true about the IP of a machine

a) It is 48-bit and will always be unique around the world.

b) It is 48-bit and is not necessarily be unique around the world.

c) It is 32-bit and will always be unique around the world.

d) It is 32-bit and is not necessarily be unique around the world.

Ans. c

28) A machine having 64MB memory runs a executable which is 300MB on disk. This is achieved by:

a) Use of FAR pointers

b) Page swapping.

c) Save some variables on another machine on network.

d.) Cannot be run on the machine.

Ans. b

29) Which of the following is true about thread and process startup speed:

a.) The startup of a thread is faster than a process.

b.) The process startup is faster as it is directly controlled by the OS.

c.) They will be equal.

d.) Depends on OS that is used. Faster on Windows98 slower on NT.

Ans. a

30) What causes "Thrashing" of a program :

a.) The constant swapping of program due to page faults.

b.) The inability of a program to get assess to a network resource.

c.) A near overflow / underflow of a variable.

d.) Assessing a memory area not allocated to the process.

Ans. a

31) Turbo-C is a / an

a.) IDE and C compiler/linker.

b) C-compiler/linker

c) C .

d) code generator.

Ans. a

32) The path of creation of an executable is :

a.) coding, linking, compiling, parsing.

b.) coding, parsing, compiling, linking.

c.) coding, compiling, parsing, linking.

d.) coding, compiling, linking, parsing.

Ans. b

33) Memory leak in software is due to

a.) Heavy recursion used in logic.

b.) Using structures of large size

c.) Improper use of the CPU registers.

d.) Improper release of allocated memory .

Ans. d

34) A "stable sort" is different from "sort" in the following way

a) Stable sort handles multiple thread access.

b.) Stable sort maintains the order of equal entities as it was in original sequence.

c.) Stable sort will always sort using the fastest scheme available in the library.

d.) Stable sort can handle exceptional conditions like interrupts in software.

Ans. b

35) The classic way of checking whether a mathematical expression has matched paranthesis will employ the following data structure :

a.) List.

b.) Directed Graph

c.) Threaded Binary tree.

d.) Stack.

Ans. d

36) The fastest sorting algorithm for a Random set of numbers is:

a.) Quick sort

b.) Shell sort

c.) Bubble sort

d.) Double Bubble sort.

Ans. a or b, I don't know

37) Which of these items is not a form of IPC:

a.) Shared Memory

b.) Pipes.

c.) Message queues.

d.) Semaphores

Ans. b

38) The term "socket" in software refers to:

a.) The software primitive which allows access to the hardware attached to the computer.

b.) The API exposed by drivers to assess a hardware.

c.) End point of connection used to transfer data programmatically.

d.) Th primitive used by OS to get assess to the CPU for process scheduling.

Ans. c

39) Which of these statements is True:

a.) XML is a sub-set of HTML.

b.) XML is a less generic markup language standard derived from SGML.

c.) XML is HTML for Xtended Interfaces like mobile-phones.

d.) XML is a Xtension of HTML which defines new tags.

Ans. d

40) DCOM and CORBA are:

a.) Specifications which enable faster downloads on the net.

b.) Specifications that allow objects to be accessed in a location independent manner.

c.) Parallel implementations of XML by Microsoft and Sun respectively..

d.) Specifications to store objects on disk, for later retrieval.

Ans. b

41) The Process that involves monitoring and improving the software product development is:

(a) Quality Assurance

(b) Quality Control

(c) Quality Improvement

(d) None of the above

Ans.

42) An executable test that verifies a functionality of the software unit with given input and expected output is called:

(a) Test Script

(b) Test Plan

(c) Test Case

(d) All of the above

Ans. maybe c

43) A Bug in the software is, when there is

(a) Application Crash

(b) Feature Failure

(c) Loss of Data

(d) All of the above

Ans. maybe a, or probably d

44) Testing based on External Specifications without knowledge of how the system is constructed:

(a) Black Box Testing

(b) White Box Testing

(c) Stress Testing

(d) Performance Testing

(e) None of the above

Ans. a

45) Which of the following is false

(a) QA is a process defined to attain Quality Standards

(b) QC is testing of the product during its production workflow

(c) A Test case could contain many Test Pl

Ans.

(d) A Test Plan could contain many Test Script & Test Cases

Ans. c

46) A Printing Machine that transfers impressions from Flat Plate to Rubber Cylinder, thence to paper is called

(a) Laser Printer

(b) Press, Offset

(c) Digital Printer

(d) All of the above

Ans. maybe Offset

47) Printing process in which ink is applied to paper or board from raised portions of printing plates or type is called:

(a) Printing, Letterpress

(b) Printing, Flexographic

(c) Printing, Silk screen

(d) None of the above

Ans. maybe letterpress

48) Yellow (lemon), Magenta (coldred), Cyan (blue-green) are the three

(a) Process Colors

(b) Monitor Colors

(c) Special Colors

(d) Spot Colors

Ans.

49) Leading specifies:

(a) The space between the lines in a paragraph.

(b) The space between the base of a line to the base of the following line in a paragraph.

(c) The space between the top of the X height and the bottom of the X height of the following line in a paragraph

(d) The space between the beard of the top line and the beard of the bottom-line.

Ans.

50) Which is a typical page layout program out of the following software products:

(a) Adobe Photoshop

(b) Adobe PageMaker

(c) Macromedia FreeHand

(d) Macromedia Director

Ans. b

C++ Questions

1. What does the following do:

void afunction(int *x)

{

x=new int;

*x=12;

}

int main()

{

int v=10;

afunction(&v);

cout<}

a) Outputs 12

b) Outputs 10

c) Outputs the address of v

Ans b.

2. How long does this loop run: for(int x=0; x=3; x++)

a) Never

b) Three times

c) Forever

Ans. c

3. Which uses less memory?

a)

struct astruct

{

```
int x;

float y;

int v;

};
```

b)

```
union aunion

{

int x;

float v;

};
```

c)

```
char array[10];
```

Ans. b

4. Evaluate:

```
int fn(int v)

{

if(v==1 || v==0)

return 1;

if(v%2==0)
```

return fn(v/2)+2;

else

return fn(v-1)+3;

}

for fn(7);

a) 10

b) 11

c) 1

Ans. b

5. Which of the Standard C++ casts can be used to perform a "safe" downcast:

a) reinterpret_cast

b) dynamic_cast

c) static_cast

d) const_cast

Ans. b

6.

class professor {};

class teacher : public virtual professor {};

class researcher : public virtual professor {};

class myprofessor : public teacher, public researcher {};

Referring to the sample code above, if an object of class "myprofessor" were created, how many instances of professor will it contain?

a) 0

b) 1

c) 2

d) 3

e) 4

Ans. b , here professor will be called a virtual base class since teacher and researcher derive from it virtually. This is used in multiple inheritance as shown here. If professor was not inherited virtually then there would be 2 instances of professor in the object of myprofessor.

7.

string somestring ;

Which of the following choices will convert a standard C++ string object "somestring" to a C string?

a) Copy.somestring () ;

b) somestring.c_str ()

c) &somestring [1]

d) std::cstring (somestring)

e) (char *) somestring

Ans. b

8.

class ba***

{

int x;

public:

void setx(int y) {x=y;}

};

class derived : ba*** {};

What is the access level for the member function "setx" in the class "derived" above?

a) private

b) local

c) global

d) public

e) protected

Ans. a

E) Table of Member Access Privileges

Access in Base Class Base Class Inherited as Access in Derived Class

Public Protected Private Public Public Protected No access1

Public Protected Private Protected Protected Protected No access1

Public Protected Private Private Private Private No access1

1 Unless friend declarations within the base class explicitly grant access.

So, the highest member accessibility is defined by the way a class is inherited, if it is inherited privately, then the highest member accessibility will be private. Default inheritance is private.

9.

```
class Alpha {

public:

char data[10000];

Alpha();

~Alpha();

};

class Beta {

public:

Beta() { n = 0; }

void FillData(Alpha a);


private:

int n;

};
```

How do you make the above sample code more efficient?

a) If possible, make the constructor for Beta private toreduce the overhead of public constructors.

b) Change the return type in FillData to int to negate the implicit return conversion from "int" to "void".

c) Make the destructor for Alpha virtual.

d) Make the constructor for Alpha virtual.

e) Pass a const reference to Alpha in FillData

Ans. e. since u r passing a reference hence a new array will not be created in memory, whereas if u pass by value, then an array of 10000 chars will be created. Passing by reference only creates an alias for the original parameter (i.e., it points to the original parameter) and is same as passing by address, the only difference is that it can be used like an object instead of as a pointer, i.e., if param is &a, then u will write a.member, whereas if param is *a then u will write a->member.

10.

class Foo {

int x;

public:

Foo(int I);

};

If a class does not have a copy constructor explicitly defined one will be implicitly defined for it. Referring to the sample code above, which one of the following declarations is the implicitly created copy constructor?

a) Foo(Foo *f);

b) Foo(Foo &f);

c) Foo(const Foo *f);

d) Foo(const Foo &f);

e) Foo(int);

Ans. d. copy constructor takes an arg of its own type which is passed by ref and which should not be changed hence it is const

11.

class HasStatic {

static int I;

};

Referring to the sample code above, what is the appropriate method of defining the member variable "I", and assigning it the value 10, outside of the class declaration?

a) HasStatic I = 10;

b) int static I = 10;

c) static I(10);

d) static I = 10;

e) int HasStatic::I = 10;

Ans. e

12.

class X

{

private:

int a;

protected:

X(){cout<<"X constructor was called"< ~X(){cout<<"X destructor was called"<};

Referring to the code above, which one of the following statements regarding "X" is TRUE?

a) X is an abstract class.

b) Only subclasses of X may create X objects.

c) Instances of X cannot be created.

d) X objects can only be created using the default copy constructor.

e) Only friends can create instances of X objects.

Ans. b. instances of X can be created only inside its subclasses.

13.

class Foo {

const int x;

protected:

Foo(int f);

~Foo();

};

Foo f;

Referring to the sample code above, why will the class declaration not compile?

a) The variable x is const.

b) The destructor is protected.

c) The destructor is not public.

d) The constructor is protected.

e) There is no default constructor.

Ans. e. if u don't specify a constructor for a class, then the compiler generates the default constructor for u, but if u specify a constructor apart from the default constructor, then u must give the default constructor also, but only if u r creating an object which uses the default constructor. i.e., the following code is perfectly fine

class Foo {

public:

Foo(int f){}

~Foo(){}

};

Foo f(0);

But if u try to create an object of Foo like this

Foo f;

then this will give compiler error because there is no default constructor available.

14.

class Foo {

public:

Foo(int i) { }

```
};

class Bar : virtual Foo {

public:

Bar() { }

};



Bar b;
```

Referring to the above code, when the object 'b' is defined, a compiler error will occur. What action fixes the compiler error?

a) Adding a virtual destructor to the class Bar

b) Adding a constructor to Bar which takes an int parameter

c) Adding "Foo()" to the Bar constructor

d) Adding a copy constructor to the class Foo

e) Adding "Foo(0)" to the Bar::Bar initializer list

Ans. e

15. Which one of the following describes characteristics of "protected" inheritance?

a) The base class has access only to the public or protected members of the derived class.

b) The derived class has non-public, inheritable, access to all but the private members of the base class.

c) The derived class has access to all members of the base class.

d) The private members of the base class are visible within the derived class.

e) Public members of the derived class are privately accessible from the base class.

Ans. b

16. The "virtual" specifier in a member function enables which one of the following?

a) Monmorphism

b) Late binding

c) Metamorphism

d) Solomorphism

e) Inheritance

Ans. b. consider a base class B having a virtual function Foo, and a class D derived from this class also having a function Foo. Then when u create an object of a derived class, say dobj, and a ptr to base class, say pb, and point pb to dobj by saying pb = &dobj, and then call pb->Foo then the correct version of Foo will be called, i.e., the version in D will be called becz the object is of type D. this is called late binding, i.e., deferring the decision of calling which version of Foo until runtime since at compile time the type of object to which pb points may not be known.

class B {public: virtual void Foo(){cout<<"base";}};

class D : public B {public: virtual void Foo(){cout<<"derv";}};

D dobj;

B *pb;

pb = &dobj;

pb->Foo();

the output will be 'derv'.

If Foo was not declared virtual in base class B , then the output would have been 'base'.

17.

class X

{

public:

int x;

static void f(int z);

};

void X::f(int y) {x=y;}

What is the error in the sample code above?

a) The class X does not have any protected members.

b) The static member function f() accesses the non-static z.

c) The static member function f() accesses the non-static x.

d) The member function f() must return a value.

e) The class X does not have any private members.

Ans. c

18.

template class Obj {

T my_t;

X my_x;

public:

Obj(T t, X x) : my_t(t), my_x(x) { }

};

Referring to the sample code above, which one of the following is a valid conversion operator for the type T?

a) T operator T () { return my_t; }

b) T operator(T) const { return my_t; }

c) operator(T) { return my_t; }

d) T operator T (const Obj &obj) { return obj.my_t; }

e) operator T () const { return my_t; }

Ans. e.

this will be used for casting objects of type Obj to type T

class Myclass{};

template class Obj {

T my_t;

X my_x;

public:

Obj(T t, X x) : my_t(t), my_x(x) { }

operator T () const { return my_t; }

operator X () const { return my_x; }

};

```
void main()

{

Myclass mt;

Obj myobj(10,mt);

int x = (int) myobj;

cout << x;

Myclass mobj = (Myclass) myobj;

}
```

19.

```
catch(exception &e)

{

. . .

}
```

Referring to the sample code above, which one of the following lines of code produces a written description of the type of exception that "e" refers to?

a) cout << e.type();

b) cout << e.name();

c) cout << typeid(e).name();

d) cout << e.what();

e) cout << e;

Ans. c

20.

```
int f() {
int I = 12;
int &r = I;
r += r / 4;
int *p = &r;
*p += r;
return I;
}
```

Referring to the sample code above, what is the return value of the function "f()"?

a) 12

b) 15

c) 24

d) 17

e) 30

Ans. e

Answers:

1. b

2. c

3. b

4. b

5. b

6. b

7. b

8. a

9. e

10. d

11. e

12. b

13. e

14. e

15. b

16. b

17. c

18. e

19. c

20. e

C – Questions

1. What is the output of the following code

main()

{

printf("Hello %d",printf("QUARK test? "));

}

a. Compile time error.

b. Hello QUARK test?

c. Run time error.

d. None of the above.

e. Quark Test ?Hello.

Ans. d. the output is QUARK test? Hello 12

This is because the evaluation of the parameters in a function call is done from right to left, becz the parameters were passed via a stack hence the first parameter (the leftmost one) is at the bottom of the stack and the rightmost parameter (if it is an expresseion, it will be evaluated before putting it on the stack) is on the top of the stack, hence while popping the parameters from the stack, the function printf (or any other function) gets them in the reverse order, i.e., from right to left, hence the statement printf("Quark test? ") gets evaluated first while pushing it as a parameter on to the stack, and then the statement printf("Hello %d") is executed. The '12' is the result of the return of printf("Quark test? ") which returns the number of characters printed.

2.) Out put of the following code is

main()

{

```
int i,j,A;

for (A = -1;A<=1; A++)

printf("%d\t",!!A);

}
```

a. 1 0 1

b. 65534 0 65534

c. -1 0 1

d. -65534 0 65534

e. None of the above

Ans. 1 0 1

3) What is the out put of the following code?

```
main()

{

int i=255;

printf("%d\t",++(i++));

}
```

a. Compilation error

b. Runtime error

c. 256

d. 0

e. None of the above

Ans. Compile Time Error 'Lvalue Required

4) What shall be the output of the following code?

```
main()
{
char i = 'a';
printf("%c \t %c \t", i ,(++i));
}
```

a. a b

b. Compile time error

c. b b

d. a a

e. 65 66

Ans. b b

5) What shall be the output of the following code?

```
main() {

int i,j;

printf("QUARK %s\n",main());

}
```

a. Compilation error.

b. Run-time error

c. Continuous scrolling Quark on the screen.

d. None of the above.

Ans. There is nothing on the screen and prog waits till the memory lasts and then out of memory run time error, so ans is b.

6) What shall be the output of the following code ?

```
#define f(x) x*x*x

main(){

printf("\n%d",f(2+2));

}
```

a. 8

b. 64

c. 10

d. 12

Ans. f(2+2) will expand to 2+2*2+2*2+2

= 2+4+4+2

= 12

7) What shall be the output of the following code ?

```
main()
{
void fun1(void *);
char a[] = "quark";
void *temp;
temp = a;
fun1(temp);}
void fun1(void *temp1 )
{
int t1 = 0;
while(*((char*)temp1+ t1++ )!='\0') {
printf("%c",*((char*)temp1 + t1));
}
}
```

a. Compilation error

b. ark

c. quark

d. uark

Ans, uark

8. What will be the out put of the following code?

```
void main()
{ int x=3;
printf("%d\t %d",x>>1, x<<3);
}
```

a. 1 and 4

b. 1 and 24

c. 1 and 27

d. None of the above

Ans. 1 and 24

This is because 3 in binary is 00000000 00000011 in two bytes (integer). Again, the right to left evaluation rule of parameters is applicable and so x<<3 gets executed first, it means left shift 3 times, but this operator does not change the value of x itself, it simply returns a value, so x retains its value after this operation has been carried out, so we get 00000000 00011000 which is 24, so 24 is pushed onto the stack, and then x>>1, right shift 1, 00000000 00000011, which is 1 in decimal, so 1 is pushed onto the stack, then printf("%d\t %d") gets executed displaying 1 24.

9. What will be the result of the following code?

```
int *x;

x =(int *) 15;
```

a. Compilation error

b. Compiles but gives a runtime error

c. Absolute location 15 in the memory space shall be assigned to pointer x;

d. Location 15 in the program space is assigned to pointer x;

e. Location 15 contains the address to an integer.

Ans d

10. Which of the following functions cannot be called from another file?

a. const void func(){ ……..}

b. extern void func(){………}

c. void func(){………}

d. static void func(){……….}

Ans. static

11. What will be the out come of the following code?

#include

int * func(){

```
static int x=0;

x++; return &x;

}

int main()

{

int * y = func();

printf("%d",(*y)++);

func();

printf("%d\n",*y);

return 0;

}
```

a. Compilation error.

b. Prints 1 and 3

c. Prints 1 and 3 but it is not good practice.

d. Prints 1 and 1

e. The code will not execute properly because y points to a variable whose life span is limited to execution of the function func();

Ans. Prints 1 and 3 but it is not a good practice

12. Referring to the above code , which of the following would be the correct

implementation for myFunc ?

char *format = "%d";

```
int main()

{

int x;

myFunc(scanf,&x);

printf("%d\n",x);

return(0);

}
```

a. void myFunc(int(*y)(const char*,…),int *x) {(*y)(format,&x);}

b. void myFunc(int(*y)(const char*,…),int *x) {(*y)(format,*x);}

c. void myFunc(int*y(const char*,…),int *x) {(*y)(format,&x);}

d. void myFunc(*(int y(const char*,…)),int *x) {(*y)(format,x);}

e. void myFunc(int(*y)(const char*,…),int *x) {(*y)(format,x);}

13. What shall be the output of the following C code?

```
void main()

{

unsigned int x= -1;

int y =0;

if(y<=x) printf("A is true\n");

if (y = =(x = -10)) printf("B is true\n");

if ((int) x>=y) printf("C is true\n");

}
```

a. A is true.

b. B is true.

c. C is true.

d. None of the above.

Ans. A is true because x contains −1, i.e., in binary it is ffff, i.e., all 1s, so being unsigned, all 1s are interpreted as the value 65535 and not as −1 (however, all 1s are interpreted as −1 if it is just an int), hence y<=x returns true.

14. In the following code what is the correct way to increment the variable ptr to

point to the next member of the array

```
union intfloat

{

int intArray[ 5];

float floatArray[ 5];


};
union intfloat arr[20];

void *ptr =arr;
```

a. ++(int*)ptr;

b. ptr = ptr+5;

c. ptr = ptr +sizeof(*ptr);

d. ptr = ptr+sizeof(intfloat.floatArray);

e. ptr = (void*)((union intfloat*)ptr +1);

Ans. e. ptr = (void*)((union intfloat*)ptr +1);

15.What shall be the output of the following program?

```
#define PRINTXYZ(x,y,z) printf (#x "=%d\t" #z "=%d\n", x, y)

void main() {
int x, y, z;
x=0; y=1; z=2;

x || ++y ||++z;
PRINTXYZ(x,y,z);

++x || ++y && ++z;
PRINTXYZ(x,y,z);

++x && ++y || ++z;
PRINTXYZ(x,y,z);
}
```

a. Compilation error.

b. Runtime error.

c.

x=0 z=2

x=1 z=3

x=2 z=4

d.

x=0 z=2

x=1 z=2

x=2 z=3

e. None of the above.

Ans. d.

16. What shall be the output of the following code ?

main()

{

printf("%d %d", sizeof(NULL), sizeof(""));

}

a. 1 and 0.

b. 0 and 1

c. 2 and 1

d. 4 and 1

e. None of the above

Ans. Depends on the machine and compiler. Actually it is the sizeof(int) and sizeof(char) as a string is stored as a char array terminated with 0, so sizeof("") gives 1, whereas sizeof("adsf") gives 5 (including the terminating 0). So in TurboC we get c as the answer, on VC we get d as the answer, so I guess e is the ans, i.e., None of the above.

17. What shall be the output of the following code?

int *check (int,int);

void main()

{int c,d;

c = check(11,29);

d= check(20,30);

printf("\nc=%u",c);

}

int * check(int i,int j)

{

int *p, *q;

p=&i;

q=&j;

if(i>=95)

return(q);

else

return(p);

}

a. 11

b. 29

c. Compilation error

d. Runtime error

e. None of the above.

Ans. e. None of the above. the statement c = check(11,29) is assigning an int ptr to an int, so c has an address of an int (which has gone out of scope, since the function returns the address of a variable which had its scope only inside the function, since the parameters were passed by value) so the value printed can be anything. Instead, if the statement was c = *(check(11,29)) then c would have the value stored at the address returned by the function, which would most probably be 11, but it cannot be guaranteed since the variable i has fallen out of scope.

18. What shall be the output of the following code?

void main()

{int a[3][2]={ 1,2,

5,7,

6,8};

printf("\n%d",((a+1)-(&a+1)));

}

a. 0

b. -16

c. -2

d. -8

e. None of the above.

Ans. –2. I haven't been able to figure this one out. a is the address of the 2-d array, here a, &a, *a all give the same value, i.e., address of the array. (a+1) gives the address of the second row, it is the same as a[1]. *(a+1) gives the address of the first cell of the second row. **(a+1) gives the value of the element stored in the first cell in the second row. (*(a+1)+1) gives the address of the second cell of the second row. *(*(a+1)+1) gives the value of the element stored in the second cell in the second row.

19.What shall be the output of the following code?

main()

{

char str1[]="Hello";

char str2[]="Hello";

if(str1= =str2&& (*(str1+6)= =*(str2+6)))

printf("\n Equal");

else

printf("\n unequal");

}

a. Equal

b. Unequal

c. Compilation error.

d. Runtime error.

e. None of the above.

Ans. b. Unequal, because the addresses of the two strings are str1 and str2 and they are different.

20. Given that sizeof(int) is 2 , what is the output of the following code

main()

{

int a, b=255,c=127;

a=~b;

c=c^(~a & b|0);\

c=c^(~(~b));

printf("%d\n",c);

}

a. Error because of overflow;

b. 255

c. –256

d. 127

e. None of the above

Ans. d. 127

ANSWERS

1. D

2. A

3. A

4. C

5. B

6. D

7. D

8. B

9. A

10. D

11. C

12. E

13. A

14. E

15. D

16. E

17. C

18. C

19. B

20. D

Q1. __stdcall calling convention

Ans. stack is cleaned by the callee

Q2. mutable key word is used for

Ans. if used with a member variable it can be changes by a function declared as a constant

Q3. foo & const ref

Ans. this is not a good practise since references are always constants

Q4. private inheritance means

Ans Non private members of a base class become private members of derived class

Q5. int I=2;

Int j=3;

Int k=4;

Cout<<(I

Ans. 1

Q6 #defines macro(a) ((a++) + (++a) + (a++))

Main()

{

cout << macro(1);

}

Ans. 4

Q7 string s = "abcd"

Char &c=s[1];

String t;

T=s;

c='z';

if(T[1]=='z')

{

cout<<"Quark";

}

else

{

cout<<"express";

}

Ans. express

Q8. int func(int I)

```
{

static int k=0;

k++;

if( K>5)

return 1;

else

return func(I-1);

}


int main()

{

cout << func(1);

}
```

Ans. 1

Q9. class base

```
{

public:

virtual func()

{

cout <<"BASE";

}
```

```
};
class der1: public base
{
public:
virtual func()
{
cout << "DER1";
}
};
class der2: public base
{
public:
virtual func()
{
cout << "DER2";
}
};
class der3: public base
{
public:
virtual func1()
{
cout << "DER3";
}
}
```

```
};

main()

{

base *pb;

pb= new der1;

pb->func();

pb= new der2;

pb->func();

pb= new der3;

pb->func();

}
```

Ans: DER1DER2BASE

Q10 Which of following will cause application crash…there are three options

Ans. char * p = new char;

Char *q=p;

Delete p;

Delete q;

Q11 for typeid to give correct result which condition is must

Ans. Class should be polymorphic

Q12 following things are default when class is created

Ans. Default constructor, copy constructor, assignment operator

Q13. which casting is done at run time

Ans. dynamic_cast