CH 13 QUIZ

Terms in this set (51)

[1301] Which line below points ppi to pi?
 	
int main()
{
double pi = 3.14159;
double *ppi;
// code goes here
// code goes here
}
	ppi = π

[1302] Assume that ppi correctly points to pi. Which line prints the value stored inside pi?
 	
int main()
{
double pi = 3.14159;
double *ppi;
// code goes here
// code goes here
}
	cout << π
	cout << ppi;
	cout << &ppi;
	cout << *pi;
	
	→ None of these

[1303] Assume that ppi correctly points to pi. Which line prints the value stored inside pi?
 	
int main()
{
double pi = 3.14159;
double *ppi;
// code goes here
// code goes here
}
	cout << *ppi;

[1304] Assume that ppi correctly points to pi. Which line prints the address of ppi?
 	
int main()
{
double pi = 3.14159;
double *ppi;
// code goes here
// code goes here
}
	cout << &ppi;

[1305] Assume that ppi correctly points to pi. Which line prints the size (in bytes) of pi?
 	
int main()
{
double pi = 3.14159;
double *ppi;
// code goes here
// code goes here
}
	cout << sizeof(*ppi);

[1306] The value for the variable a is stored:
 	
int a = 1;
void f(int b)
{
int c = 3;
static int d = 4;
}
	in the static storage area

int a = 1;
void f(int b)
{
int c = 3;
static int d = 4;
}	

[1308] The value for the variable c is stored:
 	
int a = 1;
void f(int b)
{
int c = 3;
static int d = 4;
}
	on the stack

[1309] The value for the variable d is stored:
 	
int a = 1;
void f(int b)
{
int c = 3;
static int d = 4;
}
	in the static storage area

[1310] The variable buf is a pointer to a region of memory storing contiguous int values. (This is similar to your homework, where you had a region of memory storing unsigned char values) The four lines shown here are legal. Which operation is illegal?
 	
int *p1 = buf;
const int *p2 = buf;
int * const p3 = buf;
const int * p4 const = buf;
 	
p2++;
*p1 = 3;
*p3 = 5;
p1++;
*p2 = 7
	*p2 = 7;

[1311] The variable buf is a pointer to a region of memory storing contiguous int values. (This is similar to your homework, where you had a region of memory storing unsigned char values.) The four lines shown here are legal. Which operation is illegal?
 	
int *p1 = buf;
const int *p2 = buf;
int * const p3 = buf;
const int * p4 const = buf;
	p3++;

[1312] The variable buf is a pointer to a region of memory storing contiguous int values. (This is similar to your homework, where you had a region of memory storing unsigned char values.) The four lines shown here are legal. Which operation is legal?
 	
int *p1 = buf;
const int *p2 = buf;
int * const p3 = buf;
const int * p4 const = buf;
	*p3 = 5;

[1313] These pointer should point to "nothing". Which is not correctly initialized?	vector<int> *vp;

[1314] These pointer should point to "nothing". Which is not correctly initialized?
 	
Star *ps = NULL;
 	
vector<int> *vp(0);
 	
int *pi = nullptr;
 	
double *pd{};
 	
All are correctly initialized to point to nothing
	All are correctly initialized to point to nothing

[1315] Which of these is the preferred way to initialize a pointer so that it points to "nothing"?	int *pi = nullptr;
[1317] All of these are legal C++ statements; which of them uses the C++ address operator?	int *p = &a;

int a = 3, b = 4;	
[1319] All of these are legal C++ statements; which of them uses the C++ pointer declarator? int a = 3, b = 4;	int *p = &b;
[1320] All of these are legal C++ statements; which of them uses the C++ dereferencing operator? int a = 3, b = 4;	int x = *p;
[1321] All of these are legal C++ statements; which of them uses indirection? int a = 3, b = 4;	int x = *p;
[1322] In C++, global variables are stored:	in the static storage area
[1323] What is true about an uninitialized pointer?	Dereferencing it is undefined behavior

[1324] What is true about this code? int n{500}; int *p = &n;	*p is the value of n
[1325] What is true about this code? int * choice;	choice contains an undefined address
[1326] How can we print the address where n is located in memory? int n{500};	cout << &n << endl;

[1327] Which expression obtains the value that p points to?
 	
int x(100);
int *p = &x;
	*p

[1328] What is a common pointer error?	Using a pointer without first initializing it
[1329] What is printed when you run this code? int x(100); cout << &x << endl;	The memory location where x is stored

[1330] What is printed when you run this code?
 	
int n{};
int *p = &n;
*p = 10;
n = 20;
cout << *p << endl;
	20

[1331] What is printed when you run this code?
 	
int num = 0;
int *ptr = #
num = 5;
*ptr += 5;
cout << num << " " << *ptr << endl;
	10 10

[1332] What is printed when you run this code? int *n{nullptr}; cout << n << endl;	The address value 0
[1333] What is printed when you run this code? int *n{nullptr}; cout << *n << endl;	No compilation errors, but undefined behavior

[1334] What is printed when you run this code?	The address value where n is stored

int *p = &0; cout << *p << endl;	
[1336] What is printed when you run this code? int n{}; int *p; *p = &n; cout << *p << endl;	Will not compile
[1337] What is printed when you run this code? int n{}; int *p; *p = n; cout << *p << endl;	No compilation errors, but undefined behavior when run
[1338] What is the term used to describe a variable with stores a memory address?	pointer
[1339] Which of these is not one of the three characteristics of every variable?	alias
[1340] Which area of memory is your program code stored in?	Text
[1341] Which area of memory are local variables stored in?	Stack
[1342] Which area of memory are global variables stored in?	Static storage area

[1343] Examine the following code. What is stored in c after it runs.
 	
int f(int * p, int x)
{
***p = x** * 2;
return x / 2;
}
. . .
int a = 3, b, c;
c = f(&b, a);
	1

[1344] Examine the following code. What is stored in b after it runs.
 	
int f(int * p, int x)
{
***p = x** * 2;
return x / 2;
}
. . .
int a = 3, b, c;
c = f(&b, a);
	6

[1345] Examine the following code. What is stored in a after it runs.
 	
int f(int * p, int x)
{
***p = x** * 2;
return x / 2;
}
. . .
int a = 3, b, c;
c = f(&b, a);
	3

[1346] Examine this version of the swap() function, which is different than the two versions appearing in your text. How do you call it?
 	
void swap(int& x, int * y)
{
. . .
}
. . .
int a = 3, b = 7;
// What goes here ?
	swap(a, &b);

[1347] Examine this version of the swap() function, which is different than the two versions appearing in your text. How do you call it?
 	
void swap(int * x, int & y)
{
. . .
}
. . .
int a = 3, b = 7;
// What goes here ?
	swap(&a, b);

integers?	
[1349] Assume that p1 is a pointer to an integer and p2 is a pointer to a second integer. Both integers appear inside a large contiguous sequence in memory, with p2 storing a larger address. How many total integers are there in the slice between p1 and p2?	p2 - p1;
[1350] Here is the pseudocode for the greenScreen() function in H12. What single statement sets the red, green and blue components to 0? Let p point the beginning of the image Set end to point just past the end While p != end If *(p + 3) is 0 (transparent) Clear all of the fields Increment p by 4	***(p) =** ***(p + 1) =** *(p + 2) = 0;
[1351] Here is a fragment of pseudocode for the negative() function in H12. What statement represents the underlined portion of code? Let p point to beginning of the image Let end be pixel one past the end of the image While p != end Invert the red component Move p to next component	p++;

Used to access the data inside a variable -> variable name	variable name
Determines the amount of memory required and the operations permitted on a variable -> variable type	variable type
The meaning assigned to a set of bits stored at a memory location -> variable value	variable value
An object whose value is an address in memory -> pointer	pointer
Expression using the address operator -> p = &a;	p = &a;
Expression using the reference declarator -> int x = 3;	int x = 3;
Expression using the dereferencing operator -> y = *a;	y = *a;
Expression using the pointer declarator -> double * v;	double * v;
Expression returning the number of allocated bytes used by an object -> sizeof(Star)	sizeof(Star)
Address value 0 -> nullptr	nullptr