

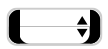
Quiz

Note: It is recommended that you save your response as you complete each question.

Question 1 (10 points)



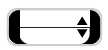
Matching the following terms with the definitions given below:



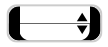
The area of memory used for allocation and deallocation of dynamic data



A simple type that only be initialized with the address of a variable



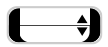
a member function invoked when an object goes out of scope



Variable created using the new operation



A pointer that points to a deallocated object



An object that has been allocated but has no pointer pointing to it



A simple type that can be assigned the address of a variable



The loss of available space that occurs when dynamic data is not deallocated properly



Accessing a variable using an address stored in a pointer



Accessing a variable using its name

1. Pointer type

2. Indirect addressing

3. Direct addressing

4. Reference type

5. Heap

6. Memory leak

7. Inaccessible object

8. Dangling pointer

9. Dynamic data

10. Destructor

Save

Question 2 (10 points)



1. Explain what is the effect of each of the following statements?

(a) `intPtr p` _____

(b) `p = new int [10];` _____

(c) `p[3] = 4;` _____

(d) `delete [] p;` _____



Save

Question 3 (20 points)



1. What is the output of the following program segment?

```
typedef int * intPtr;

intPtr p, q;
int x=3, y=6;

p=&y;
q=p;
*q = *p+1;
x=*q;
y= x+1;

cout << x << " " << y << " " << *p << " " << *q << endl;

p=new int;
*p = 5;
*q = *p + 2;
x = *q;

cout << x << " " << y << " " << *p << " " << *q << endl;

q=new int;
delete p;
p=q;
delete p;
p = NULL;
q = NULL;

cout << x << " " << y << " " << *p << " " << *q << endl;
```



Save

Question 4 (10 points)



Given the following pointer variables:

```
int x;  
int* p;  
p = &x;
```

Which of the following code segments will cause segmentation fault?

☐ *p = 100.12;

☐ p = NULL;
cout << *p;

☐ p = NULL;
cout << p;

☐ p = NULL;
cout << &p;

Save

Question 5 (10 points)



Assume we have the following structure definition:

```
struct Student  
{  
    string name;  
    string email;  
    string major;  
};
```

Which of the following statements declares a pointer to a Student struct?

☐ Student aPointerToStudent;

☐ *Student p;

☐ Student p*;

☐ Student *p;

Save

Question 6 (10 points)



Given the following structure and variables:

```
struct Student {  
    string name;  
    string email;  
};  
  
Student john;  
  
Student* p;
```

Which of the following statements makes the pointer p point to the variable john?

☐ p = *john;

☐ p = &john;

☐ &p = john;

☐ *p = &john;

☐ *p = john;

Save

Question 7 (10 points)



Suppose the following structure and variables are declared:

```
struct Student {  
    string name;  
    string email;  
};  
  
Student john;  
  
Student* p = &john;
```

Which of the following statements assigns a new email to the student john?

- ☐ `p[email] = "john@mtsu.edu";`
- ☐ `p.email = "john@mtsu.edu";`
- ☐ `john[email] = "john@mtsu.edu";`
- ☐ `john->email = "john@mtsu.edu";`
- ☐ `p->email = "john@mtsu.edu";`

Save

Question 8 (10 points)



Which of the following code segments correctly declares a structure and a pointer variable in which the structure only contains the name and email?

☐ struct Student {
 string name;
 string email;
};
Student* pointer;

Student* pointer;

☐ struct Student {
 string name;
 string email;
};

☐ struct Student {
 string name;
 string email;
 Student* pointer;
};

☐ struct Student {
 string name;
 string email;
};
Student pointer;

Save

Question 9 (10 points)



Suppose the following structure and variables are declared:

```
struct Student {  
    string name;  
    string email;  
};  
  
Student john;  
Student* p = &john;
```

Which of the following statements print the student john's email?

- ☐ `cout << p.email;`
- ☐ `cout << p[email];`
- ☐ `cout << john->email;`
- ☐ `cout << p->email;`
- ☐ `cout << john[email];`

Save

Save All Responses

Go to Submit Quiz