

```
1. Assume these declarations: char c = 'T', d = 'S';
                               char* p1 = &c;
                               char* p2 = &d;
                               char* p3;
```

```
Assume the output of the following is as shown: cout << &c;           6940
                                                cout << &d;           9772
```

What will be printed when the following statements are executed, one right after the other?

```
p3 = &d;
cout << "*p3 = " << *p3 << endl;   Output: *p3 = S

p3 = p1;
cout << "*p3 = " << *p3           Output: *p3 = T
    << ", p3 = " << p3 << endl;   Output: , p3 = 6940

*p1 = *p2;
cout << "*p1 = " << *p1           Output: *p1 = S
    << ", p1 = " << p1 << endl; Output: p1 = 6940
```

```
2. Consider these declarations and statements: int* p;
                                               int i = 42, k = 75;
                                               p = &i;
```

After the above, which of the following will change the value of i to 75?

- (A) k = i;
- (B) \*i = \*k;
- (C) p = 75;            **Answer: D**
- (D) \*p = 75;
- (E) \*i = 75;

```
3. Explain the error: char c = 'A';
                       double *p = &c;
```

**type mismatch:      &c is pointer to char**  
**p is pointer to double**

```
4. int n, sum = 0;
   for( n = 1; n <= 10; n = n + 1 )
       sum = sum + n;
```

Rewrite the above for() loop correctly using only the pointer declarations given below. In other words, don't use **n**, use the pointer to **n**, and don't use **sum**, use the pointer to **sum**:

```
int n, *pn = &n;
int sum = 0, *psum = &sum;
// for loop rewritten using only the pointer variables:
for( *pn = 1; *pn <= 10; *pn = *pn + 1 )
    *psum = *psum + *pn;
```