Tutorial 4
NWEN241
Systems Programming – Quiz 1 and Pointers

Kirita-Rose Escott

kirita-rose.escott@ecs.vuw.ac.nz

Content

- Quiz 1
    - Solutions
    - Inheritance Solutions
- Pointers
    - Introduction to pointers

Quiz 1

In C++, the expression `float(7/2)` will evaluate to `3.5`

ANSWER:

- False – the compiler will compute 7/2 first, where they are evaluated as whole integers (7/2 = 3) before converting the answer to a float: 3.0

Quiz 1

Which of the following is a valid C/C++ identifier?

- `1node_Counter`
- `$value`
- `static`
- `_counter_variable2`

ANSWER:

- `_counter_variable2` - identifiers cannot start with a number or a $ symbol, static is a keyword

Quiz 1

Consider the following function-like macro:

```
        #define MACRO_ME(X,Y)      X*Y
```

To what value does the macro evaluate when invoked as `MACRO_ME(1+2,4-3)`?

ANSWER:

1 + 2 * 4 – 3

1 + 8 – 3

9 – 3

= 6

Quiz 1

What value is assigned to `j` in the expression `j = ++i % i – 2` when `i = 3`?

ANSWER:

- -2
  - `++i` increments `i` to be 4, which means :
    - `i % i` is `4 % 4 = 0 - 2 = -2`

Quiz 1

Consider the following statement:

```
    char str[12] = “Twelve\0ab”;
```

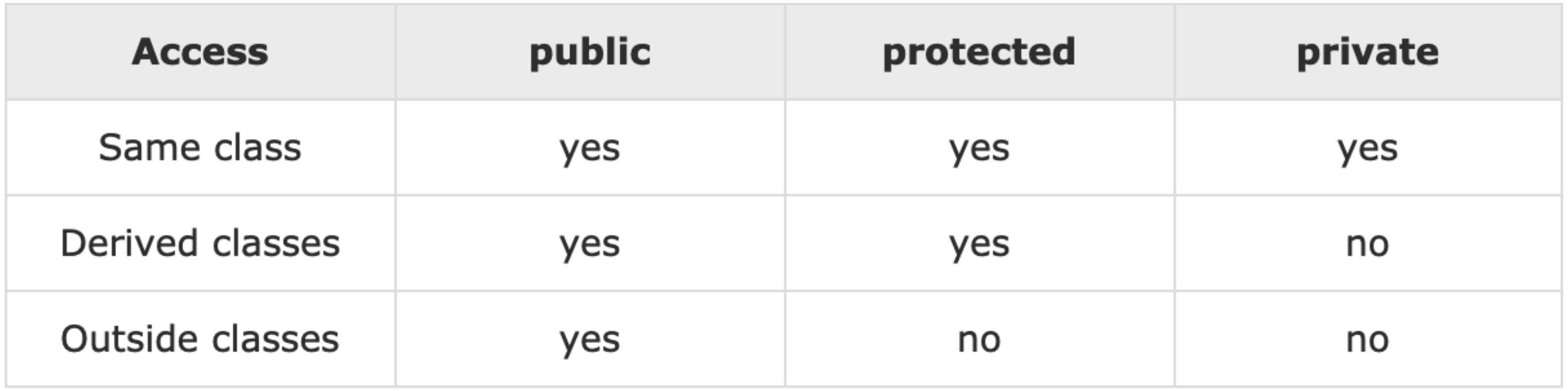
What is the length of the string?

ANSWER:

- `6 –` the `\0` terminates the string so only “Twelve” is stored in the string, which is length equal to 6

C++ Inheritance

- Inheritance allows us to define a class in terms of another class, which makes it easier to create and maintain an application
- Base class and derived class
  - class derived-class: access-specifier base-class
- A derived class can access all the non-private members of its base class



Access	public	protected	private
Same class	yes	yes	yes
Derived classes	yes	yes	no
Outside classes	yes	no	no

Quiz 1 Inheritance

Consider the following class declaration:

```
class A {
public:
        void f1(int x) { a = x; }
        void f2(char) const;
        A(int);
        A(int, int, int);
protected:
        A();
private:
        int a;
        int b;
        int c;
 };
```

Quiz 1 Inheritance Continued..

Select all statements that are true about class A.

- It is possible to create an instance of the class using the default constructor, such as in the declaration `A a;`
- The member function `f2()` can modify the member variables a, b and c.
- The member function `f1()` may be compiled inline.
- The member variables a, b, and c can be assigned values directly from outside the class.
- The following syntax for implementing the constructor that takes in 1 parameter is correct:

```
void A::A(int x) {
        a = x;
}
```

- The following syntax for implementing the constructor that takes in 3 parameters is correct:

```
void A::A(int x, int y, int z) : a(x), b(y), c(z) { }
```

Quiz 1 Inheritance Continued..

ANSWER:

- The member function `f1()` may be compiled inline because the body of the function has been provided inside the class:

```
  void f1(int x) { a = x; }
```

Quiz 1 Inheritance

Consider the following code snippet:

```
class A {
public:
        int f1(void) { aa = 0; }
        int f2(void) { aaa = 0; }
protected:
        int aa;
private:
        int aaa;
};

class B: protected A {
public:
        int f3(void) const;
};
```

Quiz 1 Inheritance Continued..

Select all statements that are true about classes A and B.

- A is a subclass of B.
- B has no access to member variable aaa.
- B has access to member variable aa.
- It is possible to invoke the member functions f1() and f2() on instances of B, for example :

```
B b;
b.f1(); // Invoke f1()
b.f2(); // Invoke f2()
```

- Calling the member functions f1() and f2() from within member function f3() is legal.
- It is not possible to instantiate either A or B because there is no default constructor.

Quiz 1 Inheritance Continued..

ANSWER:

- B has no access to member variable `aaa`.
- B has access to member variable `aa`.

Quiz 1 Inheritance

Consider the following code snippet:

```
class A {
public:
        int f1(void) { aa = 0; }
        virtual int f2(void) = 0;
protected:
        int aa;
private:
        int aaa;
};

class B: public A {
public:
        int f3(void) const;
};
```

Quiz 1 Inheritance Continued

Select all statements that are true about classes A and B.

- A is an abstract class
- B is an abstract class
- B has access to member variable `aaa`.
- It is possible to instantiate B and this will invoke the default constructor.
- The member function `f2()` is a pure virtual function.

Quiz 1 Inheritance Continued..

ANSWER:

- A is an abstract class
- B is an abstract class
- The member function `f2()` is a pure virtual function.

How ”big” is a pointer variable?

```
#include <stdio.h>

int main(void)
{
		char *cp;
		int *ip;
		float *fp;
		double *dp;

		printf("sizeof(cp): %d\n", sizeof(cp));
		printf("sizeof(ip): %d\n", sizeof(ip));
		printf("sizeof(fp): %d\n", sizeof(fp));
		printf("sizeof(dp): %d\n", sizeof(dp));

		return 0;
 }
```

Will the output be the same always?

What is the initial value of a pointer variable?

```
#include <stdio.h>

int main(void)
{
	int *ip;
	printf("ip (in decimal): %u\n", (unsigned long)ip);
	printf("ip (in hex) : 0x%X\n", (unsigned long)ip);

	return 0;
}
```

Will the output be the same always?

Assigning an address to a pointer variable

```
#include <stdio.h>

int main(void)
{
	int x;
	int *ip = &x;

	printf("ip : 0x%X\n", (unsigned long)ip);
	printf("&x : 0x%X\n", (unsigned long)&x);

	return 0;
}
```

Will the output be the same always?

Indirection

```
#include <stdio.h>

int main(void)
{
        int x = 123456;
        int *ip = &x;

        printf("ip : 0x%X\n", (unsigned long)ip);
        printf("*ip : %d\n", *ip);

return 0;
}
```

Arrays and pointers

```
#include <stdio.h>

int main(void)
{
        int x[] = {1, 2, 3, 4, 5, 6, 7, 8};
        int *ip = x;

        printf("ip   : 0x%X\n", (unsigned long)ip);
        printf("x    : 0x%X\n", (unsigned long)x);
        printf("&x[0]: 0x%X\n", (unsigned long)&x[0]);

        return 0;
}
```

Why are ip, x, and &x[0] the same?

Arrays and pointers

```
#include <stdio.h>

int main(void)
{
	int x[] = {1, 2, 3, 4, 5, 6, 7, 8};
	int *ip = x;

	printf("ip          : 0x%X\n", (unsigned long)ip);
	printf("sizeof(int) : %d\n", sizeof(int));
	printf("ip+1        : 0x%X\n", (unsigned long)
(ip+1));

	return 0;
}
```

Pointers going haywire

```
#include <stdio.h>

int main(void)
{
        int x[] = {1, 2, 3, 4, 5, 6, 7, 8};
        int *ip = x;

        printf("ip              : 0x%X\n", (unsigned long)ip);
        printf("*(ip-1)         : %d\n", *(ip-1));

        return 0;
}
```

Iterating over an array with pointers

```
#include <stdio.h>

int main(void)
{
        int x[] = {1, 2, 3, 4, 5, 6, 7, 8};
        int *ip = x;

        for(; ip < x + 8; ip++) {
                printf("ip  : 0x%X\n", (unsigned long)ip);
                printf("*ip : %d\n\n", *ip);
        }

        return 0;
}
```

Next Week’s Tutorial

- Will be setting the tutorial up to solve two problems
  - Standard C Problem
    - A function which counts the number of whitespace characters
  - Standard C++ Problem
    - A class with only one static method to count the number of digits