Quiz #2

[CS 225] Advanced C/C++

DigiPen INSTITUTE OF TECHNOLOGY SINGAPORE

GOH WEI ZHE	weizhe.goh	B	05/11/20	F
Name	Student login	Section	Date	Grade

Notes: This is a short-answer, closed books quiz. Do not collaborate or copy other people's work. Please write legibly – I can give points only for correct, clear answers I am able to read.

1. Indicate truthfulness of statements

a)	A bit is the smallest addressable unit of data storage in memory of modern computer systems and personal computers.	F ✓
b)	A byte is always defined as a group of 8 bits.	T ✗
c)	A mask is one or more bytes with predefined sequence of control bits for selecting which bits of an input to include or omit in a bitwise operation.	T ✓
d)	Bitwise NOT of an integer value returns an integer with a value 1 when it was 0; otherwise returns 0.	T ✗
e)	Bitwise OR can be used for selective setting of bits indicated by a mask in an input value.	T ✓
f)	An integer after a bitwise SHIFT LEFT by 2 places will always have 0s present at its two least significant bits.	T ✓
g)	An integer after a bitwise SHIFT RIGHT by 4 places will always have 0s present at its four most significant bits.	T ✗
h)	A hexadecimal (base-16) integer literal value in code has a prefix 0x (i.e. 0x123F), while binary (base-2) has a prefix 0b (i.e. 0b1010)	T ✓

return static_cast<unsigned short>(val & ~(1<<14));

2. Define functions

a) Define a function `clear14(val)` that accepts `unsigned short` and returns `unsigned short` representing `val` with the 2nd most significant bit cleared. Then demonstrate the calculations it performs for a call: `clear14(0x413E);`

Code	Calculations

#include <bitset>
unsigned short clear14 (unsigned short val)
{
std::bitset<8> unsigned short b;
b[6] = 0;
b[7] = 0;
return val = val & ~b;
}
return static_cast<unsigned short>(value & ~(1<<14));
	25
	16 | 413
	-32
	93
	-80
	13
	16 32 48 64 80 96
	9 A B C D

b) Define a function `toggleN(val, pos)` that accepts two `unsigned char` values and returns `unsigned char` representing `val` with a bit at the position `pos` (counted from 0) toggled. Demonstrate the calculations for a call: `toggleN(045, 6);`

Code	Calculations

#include <bitset>
unsigned char toggleN (unsigned char val, unsigned char pos)
{
std::bitset<8> unsigned char b;
b[pos] = 1;
return val = val ^ b;
}
return static_cast<unsigned char>(val ^ (1 << pos));
	√45
	45
	-32
	13
	-8
	5
	-4
	1
	-1
	0
	0 1 2 3 8 14 22

End of quiz.

0x413E: 0100 0001 0011 1110 & 1011 1111 1111 1111 = 0000 0001 0011 1110 → 0x13E

045: 0010 0101 & 0100 0000 = 0110 0101 → 145

00101101

value | mask = setting

value & ~mask = clearing

value ^ mask = negating