Exe 1 (PG17)

1a) Invalid. Identifier should not begin with a number.

b) Valid. Identifier can consist of alphabet and underscore character.

c) Invalid. Embedded space should not be included in an identifier.

d) Invalid. The only symbol character that an identifier can begin is underscore character.

e) Valid. The identifier begins with an underscore character.

f) Invalid. C++ reserved word should not be used as an identifier.

g) Invalid. # symbol cannot be used as first character of an identifier.

h) Valid. “Float” is different with “float” because case sensitive differentiate them although “float” is a C++ reserved word.

i) Valid. Identifier can consist of alphabet and numeral character.

Exe 5 (PG 19)

5a) id = 109. 2 bytes.

b) symbol = ‘$’. 1 byte.

c) average = 4.5. 4 bytes.

d) population = 4567890. 4 bytes.

e) volume = 6788.987. 8 bytes.

Exe 6 (PG19)

6a) Incorrect. Assignment symbol is absent.

float number = 12;

b) Incorrect. Single quotation mark should be used instead of double quotation mark.

char letter = ‘b’;

c) Incorrect. Integer type variable cannot hold decimal number.

double mark = 99.9;

d) Incorrect. Long type variable cannot hold decimal number.

double snum = 888888.88

e) Incorrect. There are no data type present on variables w2 and w4.

double w1 = 10, w2 = 2.55, w4 = 940;

Exe 7 (PG19)

#include <iostream>

using namespace std;

int main(){

char name[20], birth\_year[20], home\_town[20];

cout << "Please enter your name : ";

cin.getline(name,20);

cout << "Please enter your birth year : ";

cin.getline(birth\_year,20);

cout << "Please enter your home town : ";

cin.getline(home\_town,20);

cout << “Your name : ” << name << endl;

cout << “Your birth year : ” << birth\_year << endl;

cout << “Your hometown : ” << home\_town << endl;

}

Exe 8 (PG19)

8a, 8b, 8c, 8d and 8g.

8a) Variable m and r is integer type and double type respectively. From the assignment of m = r,

double type variable r cannot be assigned into integer type variable m.

8b) Variable m and n are integer type variables. Since operation of n-2.3 suppose is a decimal

number, thus integer type variable m cannot hold decimal numbers.

8c) Operation should be written at right hand inside instead of at left hand side, for example r = s+2.

8d) Variable m and s are integer type and double type respectively. Since 12/s is a decimal number, thus it should not be assigned into a integer type variable m.

8g) Modulus operation can only works with two integer number.

Exe 9 (PG20)

9a) int x;

b) float y;

c) char z;

d) x = 4;

e) y = x\*5.0;

f) y /= 3.5;

g) z = ‘F’;

Exe 10 (PG20)

10a) sqrt(pow(height,2)+pow(length,2));

b) 1/(1+pow(x,2))

c) pow(x,2)+3\*x+2;

d) 3.14159\*pow(r,2);

Exe 11 (PG20)

11a) 3

b) 13.6667

c) Error. Modulus operation can only work with two integer number

d) 25

Exe 12 (PG21)

12a) Higher to lower order of evaluation, () 🡪 \* /

x = 8\*(2+3)/(4-2);

x = 8\*5/(4-2);

x = 8\*5/2;

x = 40/2;

x = 20 (ans)

b) Higher to lower order of evaluation, () 🡪 / 🡪 +-

x = (12+4)/4+30/3-3;

x = 16/4+30/3-3;

x = 4+30/3-3;

x = 4+10-3;

x = 14-3;

x = 11 (ans)

c) Higher to lower order of evaluation, -- ++ 🡪 \* 🡪 % 🡪 +

x = a+b\*c-- % ++b;

x = 3+5\*4%++5;

x = 3+5\*4%6;

x = 3+20%6;

x + 3+2;

x = 5(ans)

Exe 13 (PG22)

13a) z = 0.8

b) z = 0.8

c) z = 0

d) c = 2

e) c = 2

f) c = 1

Exe 14 (PG23)

14a) Value of k changed. k = 5

b) Value of j changed. j = 4

c) Value of m changed. m = 1

d) Value of m changed. m = -2