EE2310 C++程式設計 HW2 (Ch 4,5) due: 4/11/2019

**A. Q & A: 20%**

**1.** What value will be stored in the variable t after each of the following statements executes?

A) t = (12 > 1);\_\_1\_\_\_\_\_\_\_\_

B) t = (2 < 0);\_\_\_\_0\_\_\_\_\_\_

C) t = (5 == (3 \* 2));\_\_\_0\_\_\_\_\_\_\_

D) t = (5 == 5);\_\_\_\_1\_\_\_\_\_\_

**2.** Write an if/else statement that prints “Excellent” when score is 90 or higher,“Good” when score is between 80 and 89, and “Try Harder” when score is lessthan 80.

int score;

cin >> score;

if(score >= 90 && score <=100)

cout << "Excellent" << endl;

else if(score >= 80 && score < 90)

cout << "Good" << endl;

else if(score >= 0 && score < 80)

cout << "Try Harder" << endl;

else if(score < 0 || score >100)

cout << "You enter a invalid score!Try again." << endl;

**3.** Each of the following program segments has errors. Find as many as you can.

A)

cout<< "Enter your 3 test scores and I will ";

<< "average them:";

int score1, score2, score3,

cin>> score1 >> score2 >> score3;

double average;

average = (score1 + score2 + score3) / 3.0;

if (average = 100);

perfectScore = true;// Set the flag variable

cout<< "Your average is " << average <<endl;

boolperfectScore;

if (perfectScore);

{

cout<< "Congratulations!\n";

cout<< "That's a perfect score.\n";

cout<< "You deserve a pat on the back!\n";

cout<< "Enter your 3 test scores and I will ";

cout << "average them:";

int score1, score2, score3;

cin>> score1 >> score2 >> score3;

double average;

average = (score1 + score2 + score3) / 3.0;

cout<< "Your average is " << average <<endl;

if (average == 100)

{

bool perfectScore;

perfectScore = true;// Set the flag variable

if (perfectScore)

{

cout<< "Congratulations!\n";

cout<< "That's a perfect score.\n";

cout<< "You deserve a pat on the back!\n";

}

}

B)

double num1, num2, quotient;

cout<< "Enter a number: ";

cin>> num1;

cout<< "Enter another number: ";

cin>> num2;

if (num2 == 0)

cout<< "Division by zero is not possible.\n";

cout<< "Please run the program again ";

cout<< "and enter a number besides zero.\n";

else

quotient = num1 / num2;

cout<< "The quotient of " << num1 <<

cout<< " divided by " << num2 << " is ";

cout<< quotient <<endl;

double num1, num2, quotient;

cout<< "Enter a number: ";

cin>> num1;

cout<< "Enter another number: ";

cin>> num2;

if (num2 == 0)

{

cout<< "Division by zero is not possible.\n";

cout<< "Please run the program again ";

cout<< "and enter a number besides zero.\n";

}

else

{

quotient = num1 / num2;

cout<< "The quotient of " << num1;

cout<< " divided by " << num2 << " is ";

cout<< quotient <<endl;

}

C)

inttestScore;

cout<< "Enter your test score and I will tell you\n";

cout<< "the letter grade you earned: ";

cin>>testScore;

if (testScore< 60)

cout<< "Your grade is F.\n";

else if (testScore< 70)

cout<< "Your grade is D.\n";

else if (testScore< 80)

cout<< "Your grade is C.\n";

else if (testScore< 90)

cout<< "Your grade is B.\n";

else

cout<< "That is not a valid score.\n";

else if (testScore<= 100)

cout<< "Your grade is A.\n";

int testScore;

cout<< "Enter your test score and I will tell you\n";

cout<< "the letter grade you earned: ";

cin>> testScore;

if (testScore< 60)

{

cout<< "Your grade is F.\n";

}

else if (testScore< 70)

{

cout<< "Your grade is D.\n";

}

else if (testScore< 80)

{

cout<< "Your grade is C.\n";

}

else if (testScore< 90)

{

cout<< "Your grade is B.\n";

}

else if (testScore<= 100)

{

cout<< "Your grade is A.\n";

}

else

{

cout<< "That is not a valid score.\n";

}

D)

doubletestScore;

cout<< "Enter your test score and I will tell you\n";

cout<< "the letter grade you earned: ";

cin>>testScore;

switch (testScore)

{ case (testScore< 60.0):

cout<< "Your grade is F.\n";

break;

case (testScore< 70.0):

cout<< "Your grade is D.\n";

break;

case (testScore< 80.0):

cout<< "Your grade is C.\n";

break;

case (testScore< 90.0):

cout<< "Your grade is B.\n";

break;

case (testScore<= 100.0):

cout<< "Your grade is A.\n";

break;

default: cout<< "That score isn't valid\n"; }

int testScore;

cout<< "Enter your test score and I will tell you\n";

cout<< "the letter grade you earned: ";

cin>>testScore;

if (testScore< 60.0)

{

testScore = 'F';

}

else if (testScore < 70.0)

{

testScore = 'D';

}

else if (testScore < 80.0)

{

testScore = 'C';

}

else if (testScore < 90.0)

{

testScore = 'B';

}

else if (testScore <= 100.0)

{

testScore = 'A';

}

switch (testScore)

{

case 'F':cout<< "Your grade is F.\n";

break;

case 'D':cout<< "Your grade is D.\n";

break;

case 'C':cout<< "Your grade is C.\n";

break;

case 'B':cout<< "Your grade is B.\n";

break;

case 'A':cout<< "Your grade is A.\n";

break;

default: cout<< "That score isn't valid\n";

}

**4.** Each of the programs in this section has errors. Find as many as you can.

A)

// This program displays the sum of two numbers.

#include <iostream>

using namespace std;

int main()

{

int choice, num1, num2;

do

{

cout<< "Enter a number: ";

cin>> num1;

cout<< "Enter another number: ";

cin>> num2;

cout<< "Their sum is " << (num1 + num2) <<endl;

cout<< "Do you want to do this again?\n";

cout<< "1 = yes, 0 = no\n";

cin>> choice;

} while (choice = 1)

return 0;

}

// This program displays the sum of two numbers.

#include <iostream>

using namespace std;

int main()

{

int choice, num1, num2;

do

{

cout<< "Enter a number: ";

cin>> num1;

cout<< "Enter another number: ";

cin>> num2;

cout<< "Their sum is " << (num1 + num2) <<endl;

cout<< "Do you want to do this again?\n";

cout<< "1 = yes, 0 = no\n";

cin>> choice;

} while (choice == 1);

return 0;

}

B)

// This program displays the sum of the numbers 1 - 100.

#include <iostream>

using namespace std;

int main()

{

int count = 1, total;

while (count <= 100)

total += count;

cout<< "The sum of the numbers 1 - 100 is ";

cout<< total <<endl;

return 0;

}

// This program displays the sum of the numbers 1 - 100.

#include <iostream>

using namespace std;

int main()

{

int count = 1, total = 0;

while (count <= 100)

{

total += count;

count++;

cout<< "The sum of the numbers 1 - 100 is ";

cout<< total <<endl;

}

return 0;

}

**5.** This program determines the fee for a cat or dog pet tag. It uses nested if/else statements.  
 #include <iostream>  
 #include <string>  
 using namespace std;  
 int main()  
 {  
 string pet; // "cat" or "dog"  
 char spayed; // 'y' or 'n'  
 // Get pet type and spaying information  
 cout << "Enter the pet type (cat or dog): ";  
 cin >> pet;  
 cout << "Has the pet been spayed or neutered (y/n)? ";  
 cin >> spayed;  
 // Determine the pet tag fee   
 if (pet == "cat")  
 { if (spayed == 'y')   
 cout << "Fee is $4.00 \n";  
 else  
 cout << "Fee is $8.00 \n";  
 }  
 else if (pet == "dog")  
 { if (spayed == 'y')  
 cout << "Fee is $6.00 \n";  
 else  
 cout << "Fee is $12.00 \n";  
 }  
 else  
 cout << "Only cats and dogs need pet tags. \n";  
 return 0;  
 }

Answer whether the following questions is correct or not:

Run Input data Fee Information Correct?

1 cat y Fee is $4.00\_\_\_\_\_\_\_\_ \_\_\_o\_\_

2 cat n Fee is $8.00\_\_\_\_\_\_\_\_ \_\_o\_\_\_

3 cat Y Fee is $8.00\_\_\_\_\_\_\_\_ \_\_\_x\_\_\_

4 dog y Fee is $6.00\_\_\_\_\_\_\_\_ \_\_o\_\_\_

5 dog n Fee is $12.00\_\_\_\_\_\_\_ \_\_\_o\_\_

6 dog Y Fee is $12.00\_\_\_\_\_\_\_ \_\_\_x\_\_\_

7 hamster n Only cats and dogs need pet tags. \_\_o\_

**B. True/False是非題: 10%**

**1.** Characters, string objects, and C-strings can be compared with any of the relational operators.

X

**2.** The three logical operators, AND, OR, and NOT, all have the same precedence.

x

**3.** The following C++ test checks if the variable child is in the range 3-12.

if (child >= 3 || child <= 12)

x

**4.** The following C++ test checks if the variable child is in the range 3-12.

if (child >= 3 &&<= 12)

x

**5.** The statement pass = (score >= 7) ?true : false; does exactly the same thing as the if/else statement below:

if (score >= 7)

pass = true;

else

pass = false;

o

**6.** The following statement s will not print anything.

x = 5;

if (x < 5)

cout<< "Hello ";

cout<< "world \n";

o

**7.** The if-statement’s advantage over the switch statement is increased clarity, especially for large numbers of items that must be compared. This makes it a good choice for implementing menus.

o

**8.** An initialization expression may be omitted from the for loop if no initialization is required.

o

**9.** The following program will print out countdown from 10:

x = 10;

while(x > 0);

{

cout << x << endl;

x--;

}

x

**10.** When a loop is nested inside another loop, the inner loop goes through all its iterations for each iteration of the outer loop.

o

**C. Choice選擇題: 20%**

**1.** A(n) \_\_\_\_\_\_\_\_ is a variable, usually a bool, that signals when a condition exists.

A) flag

B) identifier

C) named constant

D) condition variable

E) logical variable

A

**2.** When an if statement is placed within the conditionally-executed code of another if statement, this is known as a(n) \_\_\_\_\_\_\_\_.

A) complex if

B) overloaded if

C) nested if

D) conditional if

E) double if

C

**3.** When a program lets the user know that an invalid menu choice has been made, this is an example of \_\_\_\_\_\_\_\_.

A) input validation

B) output validation

C) menu reselection

D) invalidation

E) being user unfriendly

A

**4.** The default section of a switch statement performs a similar task as the \_\_\_\_\_\_\_\_ portion of an if/else if statement.

A) conditional test

B) break

C) trailing else

D) else if

E) body

C

**5.** What will the following code print?

num = 8;

cout<< --num<< " ";

cout<<num++ << " ";

cout<<num;

A) 7 7 8

B) 7 8 8

C) 8 7 7

D) 8 7 8

E) None of these

A

**6.** The while loop is a(n) \_\_\_\_\_\_\_\_ loop and the do-while loop is a(n) \_\_\_\_\_\_\_\_ loop.

A) finite, infinite

B) infinite, finite

C) simple, complex

D) pretest, post test

E) post test, pretest

D

**7.** The statements in the body of a do-while loop are always executed \_\_\_\_\_\_\_\_.

A) exactly once

B) at least once

C) at least twice

D) forever until the user hits the break key

E) until the test condition becomes true

B

**8.** A sentinel is a special value that \_\_\_\_\_\_\_\_.

A) is used for data validation

B) must be Boolean

C) marks the end of a list of values

D) must be a negative number

E) is all of the above

C

**9.** For data validation, it is best to use a(n) \_\_\_\_\_\_\_\_.

A) if statement

B) while loop

C) for loop

D) nested loop

E) switch statement

B

**10.** A for loop is considered a(n) \_\_\_\_\_\_\_\_ loop.

A) pretest

B) post test

C) multi test

D) sentinel controlled

E) infinite

A

**11.** What will the following expression evaluate to?

!( 6 > 7 || 3 == 4)

A) 0

B) -1

C) 6

D) true

E) false

D

**12.** Determine the values of variables in the following program and pick the right answer from the choices.

#include<iostream>

void main( )

{

int a=5,b=6,c,d,e,f,g,h;

c=a>b;

d=a<b;

e=(4!=3);

f=(8>=17);

g=(123<=123);

h=(a==b);

}

A) c=d=f, e==g=h

B) c=f=h, d=e=g

C) c=f=g, d=e=h

D) c=e=g, d=f=h

B

**13.** Two different variables in the same program may have the same name \_\_\_\_\_\_\_\_.

A) if the second one is never declared

B) if the second one is initialized with the values of the first one

C) if they always hold different values

D) if they have different scope

E) never. A program cannot have two variables with the same name

B

**14.** The ideal type of loop to use if you want a user to enter exactly 20 values is a(n) \_\_\_\_\_\_\_\_ loop.

A) do-while

B) for

C) sentinel controlled

D) infinite

E) nested

B

**15.** The \_\_\_\_\_\_\_\_ statement causes a loop to terminate early.

A) stop

B) break

C) quit

D) terminate

E) continue

B

**16.** The \_\_\_\_\_\_\_\_ statement may be used to stop a loop's current iteration and begin the next one.

A) break

B) terminate

C) next

D) continue

E) exit

D

**17.** Which of the following loop structure will not causing an infinite loop?

A) for(;;){…}

B) while(x=1){…}

C) x=0; do{…}while(x=1);

D) x=0; while(x!=0){…}

D

**18.** Which of the following statement is correct?

(i) The for loop is a pretest loop with built-in expressions for initializing, testing, and updating a counter variable. The for loop is ideal in situations where the exact number of iterations is known.

(ii) The do-while loop is a post test loop. It is ideal in situations where you always want the loop to iterate at least once. The do-while loop is a good choice for repeating a menu or for asking the user if they want to repeat a set of actions.

(iii) The while loop is a pretest loop. It is ideal in situations where you do not want the loop to iterate if the test condition is false from the beginning.

1. (i) and (iii)
2. (ii) and (iii)
3. All of them
4. None of them

C

**19.** How many total stars will be displayed by each of the following program segment?

for (row = 0; row < 10; row ++)

{ for (star = 0; star < 30; star++)

{ if (star > 10)

break;

cout << '\*';

}

cout << endl;

}

A) 100

B) 110

C) 120

D) 300

B

**20.** Which of the code segments about using file is correct? (assume the physical files used exist in disk)

A) ofstream outputFile;

string filename = "numbers.txt";

outputFile.open(filename);

B) ofstream outputFile;

outputFile.open(“payroll.dat”);

outputFile >> salary;

C) ifstream inputFile;

string filename = “data.txt”;

inputFile.open(filename.c\_str());

while(inputFile){…}

A

**D. Programming 程式題: 50%**

**1. Time Calculator: 10%**

Write a program that asks the user to enter a number of seconds.

•There are 86400 seconds in a day. If the number of seconds entered by the user is

greater than or equal to 86400, the program should display the number of days in

that many seconds.

•There are 3600 seconds in an hour. If the number of seconds entered by the user is

less than 86400, but is greater than or equal to 3600, the program should display the

number of hours in that many seconds.

•There are 60 seconds in a minute. If the number of seconds entered by the user is lessthan 3600, but is greater than or equal to 60, the program should display the numberof minutes in that many seconds.

#include <iostream>

using namespace std;

int main()

{

double second, day, hour, minute;

cout << "It is a time caculator.\n";

cout << "Please enter a number of seconds,";

cout << "and it will be transform to day,hour,minute,logically.\n";

cin >> second;

if (second >= 86400)

{

day = second / 86400;

cout << "You enter: " << second << " seconds.\n";

cout << second << " seconds eqaul to " << day << " days.\n";

}

else if (second >= 3600 && second < 86400)

{

hour = second / 3600;

cout << "You enter: " << second << " seconds.\n";

cout << second << " seconds eqaul to " << hour << " hours.\n";

}

else if (second >= 60 && second < 3600)

{

minute = second / 60;

cout << "You enter: " << second << " seconds.\n";

cout << second << " seconds eqaul to " << minute << " minutes.\n";

}

else

cout << "You enter: " << second << " seconds.\n";

return 0;

}**2. Bank Charges: 10%**

A bank charges $10 per month plus the following check fees for a commercial checking

account:

$.10 each for fewer than 20 checks

$.08 each for 20–39 checks

$.06 each for 40–59 checks

$.04 each for 60 or more checks

Write a program that asks for the number of checks written during the past month, then

computes and displays the bank’s fees for the month.

Input Validation: Do not accept a negative value for the number of checks written.

#include <iostream>

using namespace std;

int main()

{

int check;

double fee;

const int charge = 10;

cout << "please enter the number of checks written during the last months?\n";

cout << "And we will compute and display the bank's fees for the month.\n";

cin >> check;

if (check < 20 && check >= 0)

{

fee = 0.1\*check + charge;

cout << "The bank's fees for the month is " << fee << "$.\n";

}

else if (check >=20 && check <= 39)

{

fee = 0.08\*check + charge;

cout << "The bank's fees for the month is " << fee << "$.\n";

}

else if (check >=40 && check <= 59)

{

fee = 0.06\*check + charge;

cout << "The bank's fees for the month is " << fee << "$.\n";

}

else if (check >= 60)

{

fee = 0.04\*check + charge;

cout << "The bank's fees for the month is " << fee << "$.\n";

}

else

{

cout << "You enter a negative check.\n";

cout << "It is a invalid number please entered again.\n";

}

return 0;

}

**3. Arrowhead Display: 10%**

Write a program that uses nested loops to display the arrowhead pattern shown below.

+

+++

+++++

+++++++++++++

+++++

+++

+

#include <iostream>

#include <string>

using namespace std;

int main()

{

string add;

int line, number=1, middle = 13;

for(line=1;line<=7;line=line+1)

{

if(line<4 )

{

add.assign(number,'+');

number = number + 2;

cout << add << endl;

}

else if(line==4)

{

add.assign(middle,'+');

middle = 13;

cout << add << endl;

}

else

{

number = number - 2;

add.assign(number,'+');

cout << add << endl;

}

}

}

**4. The Greatest and Least of These: 10%**

Write a program with a loop that lets the user enter a series of integers, followed by 99 to

signal the end of the series. After all the numbers have been entered, the program should

display the largest and smallest numbers entered.

#include <iostream>

using namespace std;

int main()

{

int maxi, mini, number;

cout << "You can enter a number of series and we will diplay ";

cout << "the largest and the smallest number of the series.\n";

cout << "99 is the sinal that the series will be ended.\n";

cout << "You enter a number: ";

cin >> number;

if(number!=99)

{

for(maxi=mini=number;number!=99;)

{

cout << "You enter a number: ";

cin >> number;

if(number!=99)

{

if(maxi<number)

{

maxi = number;

}

else if(mini>number)

{

mini = number;

}

}

}

cout << "End of series.\n";

cout << "The largest number of you entered is "<< maxi << " .\n";

cout << "The smallest number of you entered is "<< mini < " .\n";

}

else

{

cout << "End of series.\n";

}

return 0;

}

**5. Write a program to check whether the number is prime or not: 10%**

A number is prime if it completely divisible by 1 and itself e.g., 1, 3, 5, 7, 11, 13, 17, 19, 23 etc. To check whether a number is prime or not we start from a counter c=2(every number divides by 1) continues till c<=num/2 since no number is completely divisible by a number which is more than half of that number. For example, 12 is not divisible by 7, 8, 9, 10, 11 which are more than 6.

Your program should ask user to enter a positive natural number, then output whether it is a prime or not. For example,

**Enter a positive integer number**

**17[enter]**

**The number 17 is prime**

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**int remainder, c, number;**

**cout << "Enter a positive integer number.\n";**

**cout << "We will determine whether it is prime or not.\n";**

**cout << "You enter a number: ";**

**cin >> number;**

**if(number>3)**

**{**

**for(c=2;c<=number/2;c=c+1)**

**{**

**remainder = number % c;**

**if(remainder==0)**

**break;**

**}**

**if(remainder==0)**

**{**

**cout << "The number " << number << " is not prime.\n";**

**}**

**else if(remainder!=0)**

**{**

**cout << "The number " << number << " is prime.\n";**

**}**

**}**

**else if(number<=3 && number>0)**

**{**

**if(number==2 || number==3)**

**{**

**cout << "The number " << number << " is prime.\n";**

**}**

**else if(number==1)**

**{**

**cout << "The number " << number << " is not prime.\n";**

**}**

**}**

**else if(number<=0)**

**{**

**cout << "Please enter a positive integer number.\n";**

**}**

**return 0;**

**}**