University No. | | | | | | | | | | |

The University of Hong Kong
Faculty of Engineering
Department of Computer Science
COMP1117 A/B Computer Programming
CSIS1117A/B Computer Programming I

Date: December 28, 2012 Time: 9:30am – 12:30pm

Only approved calculators as announced by the Examinations Secretary can be used in this examination. It is candidates responsibility to ensure that their calculator operates satisfactorily, and candidates must record the name and type of the calculator used on the front page of the examination script.

Brand and type of calculator:_____

Instructions:

- **Write your university number at the top of each page.**
- **Answer ALL questions in the space provided.**

Section A: True or False. (10 marks) Each correct answer will be given 1 mark. 1 mark will be deducted for each wrong answer. The minimum score for the entire section is 0 mark.

1. The integer 0 is considered true. **Ans:** [ ]
2. The opposite of "less than" is "greater than". **Ans:** [ ]
3. The body of a do-while loop always executes at least once. **Ans:** [ ]
4. Loops are used when we need our program to make a choice between two or more things. **Ans:** [ ]
5. All nested if-else statements can be converted into switch statements. **Ans:** [ ]
6. It is illegal to make function calls inside a switch statement. **Ans:** [ ]
7. A break statement in a switch stops your program. **Ans:** [ ]
8. The parameter names are mandatory in the function header. **Ans:** [ ]
9. A void function can return any value. **Ans:** [ ]
10. Friend functions may directly modify or access the private data members. **Ans:** [ ]

University No. | | | | | | | | | | |

Section B. Multiple Choices. (20 marks) There may be more than one correct choice in the questions. The answer will only be considered correct when all correct choices are written down. Each question carries 2 marks. 1 mark will be deducted for a wrong answer. The minimum score for this section is 0 mark.

1. Which of the following data types can be used in a switch controlling expression?

   (a) `int`

   (b) `char`

   (c) `string`

   (d) `double`

   Ans: [ ]

2. If you need to write a do-while loop that will ask the user to enter a number between 2 and 5 inclusive, and will keep asking until the user enters a correct number, what is the loop condition?

   (a) `(2 <= num <= 5)`

   (b) `(2 < 5 < number)`

   (c) `(2 <= number && number <= 5)`

   (d) `(2 > number || number > 5)`

   (e) `(2 > number && number > 5)`

   Ans: [ ]

3. Given the following function and local variable declarations,

```
int myInt;
float myFloat;
char myChar;
void someFunction(int& first, float second, char third);
```

   which of the following is not a correct function call?

   (a) `someFunction(1, 2.0, myChar);`

   (b) `someFunction(myInt, myFloat, myChar);`

   (c) `someFunction(myInt, 2.0, 'c');`

   (d) `someFunction(myInt, myFloat, '1');`

   Ans: [ ]

University No. | | | | | | | | | | |

4. Given the following function declarations,

```
int myFunction(int myValue);
int myArray[100];
```

which of the following is a legal call to this function?

(a) `cout << myFunction(myArray);`

(b) `cout << myFunction(myArray[0]);`

(c) `myArray = myFunction(myArray);`

(d) `myArray[1] = myFunction(myArray[0]);`

Ans: [ ]

5. Which of the following array declarations are legal?

(a) `int array[10];`

(b) `int size; cin >> size; int array[size];`

(c) `int array[] = {0,0,0};`

(d) `const int size=9; int array[size];`

Ans: [ ]

6. Which of the function declarations will accept the following two-dimension array?

```
int pages[10][30];
```

(a) `void func(int pages[][], int size);`

(b) `void func(int pages[][30], int size);`

(c) `void func(int pages[10][], int size);`

(d) `void func(int pages[10][30], int size);`

(e) `void func(int& pages, int size);`

Ans: [ ]

University No. | | | | | | | | | | |

7. Given the following code fragment,

```
int v1 = 0;
int * p1 = &v1;
int a1[10];
```

which of the following statements are not legal?

(a) `p1= a1;`

(b) `cout << p1[0];`

(c) `cin >> p1[0];`

(d) `a1 = p1;`

Ans: [ ]

8. If a program requires to dynamically allocate a two-dimensional array, you would do so by using

(a)
```
p1 = new int*[numRows];
for(int i=0; i < numRows; i++)
    p1[i] = new int[numColumns];
```

(b) `p1 = new int[numRows][numColumns];`

(c) `p1 = new[numRows][numColumns]int;`

(d) None of the above

Ans: [ ]

9. When should a member function of a class be made `private`?

(a) Always.

(b) Only if it will never be used.

(c) If it will only be used by other members of the class.

(d) Never, it is illegal to make a member function private.

Ans: [ ]

University No. | | | | | | | | | | |

10. Given the following class and object declaration,

```
class Wine
{
    public:
       Wine();
       int getAge();
       float getCost();
    private:
       int age;
       float cost;
}

Wine bottle;
```

how would you print out the age and cost of a bottle of wine?

(a) `cout << bottle;`

(b) `cout << Wine.age, Wine.cost;`

(c) `cout << bottle.getAge() << bottle.getCost();`

(d) `cout << bottle.getAge << bottle.getCost;`

(e) `cout << bottle.age << bottle.cost;`

Ans: [ ]

Section C: Written Questions (70 marks) The mark of each question is put in a pair of square bracket [ ].

1. [2] What does the following piece of code print to the screen?

```
cout << "Hello Students/n";
cout << "This is a \\" << endl;
```

Answer:

University No. | | | | | | | | | | |

2. [2] Given the following operator precedence

`!` (unary not operator) *greater than* `==, !=` (left associative) *greater than* `&&` *greater than* `||`

Write down meaning of the following expression by putting parenthesis to it.

```
x==3 || !y==5 && !z!=x
```

Answer:

---

3. [2] Given the following code, what is the final value of `i`?

```
int i,j;
for(i=0; i<4; i++)
{
    for(j=0; j<3; j++)
    {
        if(i==2) break;
    }
}
```

Answer:

---

4. [2] How many times is "Hi" printed to the screen?

```
for(int i=0; i<14 ;i++);
    cout <<"Hi\n";
```

Answer:

University No. | | | | | | | | | | |

5. [6] Correct only the *syntax errors* in the following program. Do not correct lines that do not contain any syntax error, even though they do not perform any meaningful task. On each successive line, assume that all preceding errors have been corrected. Line number is given on the left for easy reference.

```
 1: #include <iostream>
 2: #include <cmath>
 3: using namespace std;
 4: boolean is-prime(int n)
 5: {
 6:     if (n == 1) return false;
 7:     if (n == 2) return TRUE;
 8:     i=3;
 9:     while (i<=sqrt(n)) do {
10:         if ((n%i)==0) return false;
11:         i+=2;
12:     }
13:     return true;
14: }
15: int main();
16: {
17:     double num;
18:     cin << num;
19:     cout << num << " is ";
20:     if (!!is-prime(num)) cout << " not ";
21:     cout << "a prime number" << endl;
22:     return -1
23: }
```

Answer:

University No. | | | | | | | | | | |

6. [2] What is the value returned by the following function?

```
int function(void)
{
    int value = 35;
    return value++;
    value += 10;
}
```

Answer:

---

7. [4] Suppose that the input is `3 3.1 3 2.5 3 4 1.3 6 -2 5 -1`. What is the output of the following code? Assume all variables are properly declared.

```
#include <iostream>
using namespace std;

int main()
{
   int sum1=0, num, ninput;
   double val, sum2=0;

   cin >> ninput;
   cin >> val >> num;
   for (int i=0; (num > 0) && (i <= ninput); i++) {
       cin >> val >> num;
       sum1 += ++num;
       sum2 += val--;
   }
   cout << "Sum1 = " << sum1 << " Sum2 = " << sum2 << endl;
}
```

Answer:

University No. | | | | | | | | | | |

8. [2] What is wrong with the following for loop?

```
for(int i=0; i<10; i--) cout << "Hello\n";
```

Answer:

---

9. [3] Given that `array1` and `array2` are both integer arrays with number of elements specified in an integer `SIZE`, write the function body that assigns all the values in `array2` to `array1`. The function header is declared as follows.

```
void assign(int array1[], int array2[], int SIZE);
```

Answer:

---

10. [3] What is wrong with the following code fragment?

```
int *p1, *p2;
p1 = new int;
p2 = new int;
*p1=11;
*p2=0;
p2=p1;
cout << *p1 << " " << *p2 << endl;
delete p1;
delete p2;
```

Answer:

University No. | | | | | | | | | | |

11. [4] A *palindrome* is a word that reads the same either from left to right or from right to left. For example, *abcdcba* is a palindrome. Write a function to determine whether a string is a palindrome:

```
bool is_palindrome(string word)
// return true if word is a palindrome, otherwise return false
{
```

Answer:

---

12. [3] Given a linked list `list` with member function `insert(int)` to insert an integer into the beginning of the list, `remove()` to remove an integer from the beginning of the list. Consider the following operation:

```
list.insert(3), list.insert(5), list.insert(9), list.remove(),
list.remove(), list.insert(7) list.insert(11), list.insert(4)
```

Write down the result if we traverse the list and print the values from the beginning of the list to the end of the list.

Answer:

University No. | | | | | | | | | | |

13. [12] The result of a football league with 20 teams is stored in a file, named `result.dat` in the following format:

```
LFG 3 5 2 15 20
MCC 4 4 2 18 18
...
```

which contains the name of the team (one word only), followed by number of wins, draws and losses, and goal scored and goal conceded respectively.

A team receives 3 points for a win, 1 point for a draw and nothing for a loss. Team A ranks higher than team B if

- Team A has more points than team B.
- If they have the same number of points, and Team A has a larger *goal difference* (Goal Scored − Goal Conceded) than Team B.
- If they have the same point and goal difference, then they are ranked according to alphabetical order, i.e. LFG rank higher than MCC.

If the information of a team is stored in a `struct` as follows:

```
const int NTEAM=20;
struct teaminfo {
    string name;          //Name of the team
    int ngame;           //total number of games played
    int nwin;             //number of winning games
    int ndraw;            //number of draw games
    int nloss;            //number of loss games
    int goal_scored;      //number of goals scored
    int goal_conceded;    //number of goals conceded
    int goal_diff;        //goal difference
    int points;           //total points
    };
teaminfo Teams[NTEAM];
```

Write a program to

- read from `result.dat`, and stored the data into an array of struct teaminfo
- Complete the remaining entries of the struct.
- Sort the the teams and print the tables in the following formats:

```
Team  #Games   #Win   #Draw   #Loss   GS   GC   GD    Points
MCC   10       4      4       2       18   18   0     16
LFG   10       3      5       2       15   20   -5    14
```

A simple bubble sort function is given to you. You may use and modify this sorting function.

University No. | | | | | | | | | | |

```
void bsort(int x[], int n)
{
//function to sort the array x[] with n elements by bubble sort
//in descending order

    for (int i=0; i<n; i++)
        for (int j=0; j<n-i-1; j++)
           if (x[j+1]>x[j]) {
              //swap i & j
              int temp=x[j];
              x[j]=x[j+1];
              x[j+1]=temp;
           }
}
```

Answer:

University No. | | | | | | | | | | |

[continue answer for Q13]
Answer:

University No. | | | | | | | | | | |

[continue answer for Q13]
Answer:

University No. | | | | | | | | | | |

14. A class `AlarmClock` is defined as follows.

```
class AlarmClock
{
  public:
    AlarmClock(int hour, int min, int sec);
        //initialize to hour, minute and second in 24 hour format
    SetAlarm(int hour, int min, int sec); //set alarm time
    void enable(bool);        //enable or disable the alarm;
    void snooze();            //Snooze the alarm for 5 minutes;
    void tick();              //advance the clock by 1 sec, check alarm
    void show_time();         //Display the current time
    void show_alarm_time();   //Display the alarm time
    void Alarm();             //Play the alarm sound
  private:
    int hour, min, sec;       //hour, min and sec of current time
    int alm_h, alm_m, alm_s; //hour, min and sec of alarm set
    bool alarm_enable;        //enable or disable alarm
    bool snooze_on;           //snooze mode on or off
    int snztime;              //snooze time remaining in sec
};
```

(a) [2] Explain why the variables `hour`, `min` and `sec` are kept in the private section instead of public section.

Answer:

(b) [4] Write the `show_time()` member function to display the time in 12 hour format on the screen.

```
void AlarmClock::show_time()
{
   // Display the time in 12 hour format on the screen
   // e.g. 11:09:05a or _5:12:03p where _ is a blank
```

Answer:

University No. | | | | | | | | | | |

(c) [6] Implement the member function `tick()`, assuming all other member functions are available. The function will advance the time by one second. If alarm is not enabled, then do nothing. Otherwise, if snooze is true, then decrease snooze time by one second, and play alarm sound if time is up. Otherwise, check the time against the alarm time. If it is time, then play the alarm sound.

```
void AlarmClock::tick()
{
   // Move the time forward by 1 sec and check snooze, alarm
```

Answer:

University No. | | | | | | | | | | |

15. The Fibonacci number can be written as a recurrence relation as shown below:

$$f_{n+1} = f_n + f_{n-1}$$

with the first two terms defined as $f_0 = 0$ and $f_1 = 1$. The sequence obtained is $0, 1, 1, 2, 3, 5, \cdots$.

(a) [4] Write two functions to compute the Fibonacii sequence by both iteration and recursion.

```
int fib(int n)
{
// this function will return the n-th fibonacci sequence,
// we define fib(0)=0 and fib(1)=1, by recursion or iteration.
```

Answer:

University No. | | | | | | | | | | |

Answer:

(b) [2] It was found that the function does not return correct result when n becomes very large. Give an explanation, and suggest a way to circumvent it.
Answer:

(c) [2] Consider another recurrence relation given as follows:

$$g_n = 2g_{n-1} + 3$$

Write a recursive function to calculate the n-th term, assuming $g_0 = 1$.
Answer:

(d) [3] It is found that the recursive `fib()` function runs very slowly for even moderately large n, but the other function `g()` is fine. Suggest an explanation.
Answer:

— END —