

國立臺北科技大學九十六學年度碩士班招生考試

系所組別：1810 資訊工程系碩士班甲組

第三節 軟體設計 試題

第一頁 共六頁

注意事項：

1. 本試題共五題，配分共 100 分。
2. 請標明大題、子題編號作答，不必抄題。
3. 全部答案均須在答案卷之答案欄內作答，否則不予計分。

Problem 1 [9%]

Three C codes are written below. Record your answers for each of the problems in the appropriate box in the following table.

Problem	Answer
1-1	
1-2	
1-3	

註 1：請複製此答案表格於你的答案卷中。
 註 2：本題共計 3 小題，配置 9 分，每小題 3 分。

1-1. while (-- counter >= 1) { printf("%s\n", counter % 2 ? "even" : "odd"); }

The above C code can be rewritten to:

- (I) while (-- counter >= 1)
 if (counter % 2) printf("even");
 else printf("odd");
- (II) while (counter >= 1) {
 if (counter % 2) printf("even");
 else printf("odd");
 --counter;
 }
- (III) do {
 printf("%s\n", counter % 2 ? "odd" : "even");
 --counter;
 }while (counter >= 2);

- (A) I only
- (B) I and II only
- (C) II and III only
- (D) I, II, and III

```

1-2 void select( ) {
    char t, item[] = {'2', '8', '6', '1', '3'};
    int a,b,c, count=5;
    for (a=0; a < count-1; ++a) {
        c=a; t= item[a];
        for (b=a+1; b < count; ++b) {
            if (item[b] < t) c=b; t= item[b];
        }
        printf("\n a=%d, b=%d, c=%d, t=%c", a, b, c, t);
    }
}
    
```

When the above C code runs the loop at a=2 and b=4, which is true?

- (A) c=3, t=1
- (B) c=3, t=3
- (C) c=2, t=6
- (D) c=1, t=8

1-3. Which of the following codes can transfer a decimal digit to binary digit?

<p>I</p> <pre> void binary(int m) { int p=2,n=0,r=m/p; while (r>0){ r = m/p; n++; p=p*2; r = m/p; } n = p; while (n>0) { r = m/n; m = m%n; n=n/2; printf("%d",r); } } </pre>	<p>II</p> <pre> void binary (int m) { int p=2,n=0,r=m/p; while (r>0){ r = m/p; n++; p=p*2; r = m/p; } n = (p/2); while (n>0) { m = m%n; r = m/n; n=n/2; printf("%d",r); } } </pre>	<p>III</p> <pre> void binary(int m) { int n=1, r=m; while (r>0){ n=n*2; r = m/n; } n=n/2; while (n>0) { r = m/n; m = m%n; n=n/2; printf("%d",r); } } </pre>
---	---	--

- (A) I only
- (B) II only
- (C) III only
- (D) II, and III

注意：背面尚有試題

Problem 2 [26%]

A double link list is defined as a list in which each node contains exactly two links, one to the node that follows the current node and one to the node that precedes the current one. The following list-processing operations are:

- (1) insertFromFront: insert a node at the beginning of the list,
- (2) removeFromFront: delete a node at the beginning, and
- (3) display: display the list from the beginning of the list.

Please complete the following codes of statement (2-1)~(2-8), and show the output of statement (2-9) ~ (2-13) in the answer table.

Problem	Answer	Problem	Answer
2-1		2-2	
2-3		2-4	
2-5		2-6	
2-7		2-8	
2-9		2-10	
2-11		2-12	
2-13			

註 1：請複製此答案表格於你的答案卷中。
 註 2：本題共計 13 小題，配置 26 分，每小題 2 分。

```
#include <stdio.h>
#include <stdlib.h>
typedef struct dNode *DNodePtr;
typedef struct dNode{
    int data;
    DNodePtr frontLink;
    DNodePtr endLink;
}DNode;

void insertFromFront(_____, int x) {           // (2-1)
    _____;                               // (2-2)
    newNode = (DNodePtr) malloc(sizeof(DNode));
    newNode->data = x;
    newNode->frontLink=newNode->endLink=NULL;
    if ((*root) ==NULL)
        _____;                           // (2-3)
```

```
else {
    newNode->endLink = _____;           // (2-4)
    _____;                             // (2-5)
    *root = _____;                       // (2-6)
}
}

int removeFromFront(DNodePtr* root) {
    int data;
    if ((*root) == NULL)
        return -1;
    else {
        data = _____;                   // (2-7)
        *root = _____;                 // (2-8)
        return data;
    }
}

void display (DNodePtr current) {
    while (current!=NULL) {
        printf("%d ", current->data);
        current = current->endLink;
    }
}

int main(int argc, char *argv[]){
    int i,j, data[]={9,4,7,6,1,3,2,8,0,5};
    DNodePtr root = NULL;
    for (i=2,j=5;i<j;j+=3,i*=2);
    printf("%d\n", data[j-10]=i);             // (2-9)
    for (i=1; i<7; i++)
        insertFromFront(&root, data[i]);
    display(root);                            // (2-10)
    printf("\n%d",i=root->endLink->endLink->frontLink->data); // (2-11)
    for (j=i; j<7; j++)
        i=removeFromFront(&root);
    printf("\n%d\n",i);                        // (2-12)
    insertFromFront(&root, data[i--]);
    display(root);                            // (2-13)
    return 0;
}
```

Problem 3 [20%]

You have been contracted to assist a company with its payroll calculation procedure. The company employs two categories of workers: Managers and Staff. The salary and tax for both categories are calculated at different rates, but the calculations must be incorporated in one 'payroll' procedure.

Review the object-oriented program below (written in C++). Use the data provided in the program to complete the following table. You may note "N/A" (not applicable) if the statement is not meaningful.

Problem	Answer	Problem	Answer
3-1	Salary =	3-2	Tax =
3-3	Salary =	3-4	Tax =
3-5	Salary =	3-6	Tax =
3-7	Salary =	3-8	Tax =
3-9	Salary =	3-10	Tax =

註 1：請複製此答案表格於你的答案卷中。
 註 2：本題共計 10 小題，配置 20 分，每小題 2 分。

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

class Employee {
public:
    Employee(int baseSalary, int baseTax)
        :salary(baseSalary), tax(baseTax) {}

    void payroll() {
        calculateTax();
        calculateSalary();
        print();
    }
};
```

```
protected:
    int salary;
    int tax;
    virtual void calculateSalary() = 0;
    void calculateTax() { tax += salary * 0.1;}
    void print() {
        cout << "Salary = " << salary
        << " Tax = " << tax << endl;
    }
};

class Manager : public Employee{
public:
    Manager(int baseSalary, int baseTax)
        :Employee(baseSalary, baseTax){}

    void payroll(){
        calculateSalary();
        calculateTax();
        print();
    }
};

protected:
    void calculateSalary() { salary += 30000; }
    void calculateTax() { tax += salary * 0.2; }
};

class Staff : public Employee{
public:
    Staff(int baseSalary, int baseTax)
        :Employee(baseSalary, baseTax){}

protected:
    void calculateSalary() { salary += 10000; }
    void calculateTax() { tax += salary * 0.15; }
};
```

注意：背面尚有試題

```
void main()
{
    Employee* employeePtr = new Employee(6000, 500);
    employeePtr->payroll();    // Problem 3-1, 3-2
    delete employeePtr;
}
```

```
void main()
{
    Employee& employee = Manager(6000, 500);
    employee.payroll();    // Problem 3-3, 3-4
    cout << endl;
}
```

```
void main()
{
    Employee* employeePtr = new Staff(6000, 500);
    employeePtr->payroll();    // Problem 3-5, 3-6
    delete employeePtr;
}
```

```
void main()
{
    Manager* managerPtr = new Manager(6000, 500);
    managerPtr->payroll();    // Problem 3-7, 3-8
    delete managerPtr;
}
```

```
void main()
{
    Staff staff = Staff(6000, 500);
    staff.payroll();    // Problem 3-9, 3-10
}
```

注意：此處空白
後面尚有 2 題試題

注意：此處空白
後面尚有 2 題試題

注意：此處空白
後面尚有 2 題試題

Problem 4 [15%]

Five simple software construction problems are written below. Record your answers for each of the problems in the appropriate box in the following table.

Problem	Answer
4-1	
4-2	
4-3	
4-4	
4-5	

註 1：請複製此答案表格於你的答案卷中。
 註 2：本題共計 5 小題，配置 15 分，每小題 3 分。

4-1 Which of the following is NOT required of a software component?

- (A) Must be a unit of independent deployment
- (B) Must support reuse
- (C) Must be encapsulated
- (D) Must expose source code for modification

4-2 The most important characteristic of an abstract data type is:

- (A) The fact that knowledge about the implementation of an abstract data type can be used for memory optimization
- (B) The fact that all signatures are encapsulated within the abstract data type's interfaces
- (C) The separation of the implementation of the abstract data type from its use
- (D) The polymorphism that results from defining a data structure as an abstract data type

4-3 Partitioning systems into the smallest components, while maximizing reuse, will diminish which of the following:

- I. Efficiency
- II. Robustness in the face of evolution
- III. Manageability of the variety of configurations

- (A) I only
- (B) I and II only
- (C) II and III only
- (D) I, II, and III

4-4 An engineer is given the task of verifying a software release for a mission-critical system. The software is to be released on a Monday, and the verification is scheduled for completion on the following Friday. Then trouble strikes. The release will not be available until Thursday. The best course of action for the engineer is to:

- (A) Verify release criteria regardless of time line
- (B) Do whatever testing can be done by Friday
- (C) Volunteer to work over the weekend
- (D) Relax release criteria

4-5 Two different implementations of a fully tested abstract class have been produced. Members of a software engineering team need to correctly select the appropriate implementation for use. To do this the documentation for each of these implementations must include:

- I. A list of functions available and calling conventions
- II. The source code for the implementation
- III. Information about the side effects and resource usage for each function
- IV. Names of the authors of each implementation and change history

- (A) I and II only
- (B) I and III only
- (C) I, II, and III only
- (D) I, II, III, and IV

注意：背面尚有試題

Problem 5 [30%]

In this problem, we are developing a C++ program by using the **composite pattern**, which compose objects into tree structures to represent part-whole hierarchies. We have a point class and a shape class defined as follows.

```
class point {
public:
    point(int nx=0, int ny=0) {
        x = nx; y = ny;
    }
    void print() {
        cout << "(" << x << "," << y << ")";
    }
    int x, y;
};

class shape {
public:
    // Draw the shape
    virtual void Draw() = 0;
    // Calculate and return the Bounding Box of the shape
    virtual void GetBoundingBox(point *ul, point *lr)=0;
};
```

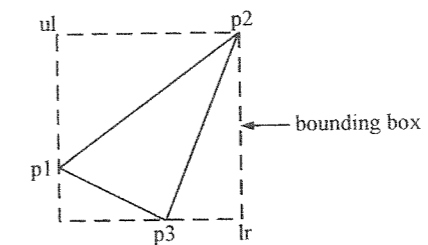
A point object records a coordinate. A shape (an abstract class) object has two member functions, Draw (draw the shape) and GetBoundingBox (find the bounding box of the shape, i.e., the **upper left**, ul, point and the **lower right**, lr, point of the shape). Suppose we have a function void draw_line(point p1, point p2) that can draw a line from point p1 to point p2, then the rectangle class, one of the shapes, can be implemented as follows.

```
class rectangle : public shape {
public:
    rectangle(point np1, point np2) {
        p1 = np1; p2 = np2;
    }
    void Draw() {
        draw_line(point(p1.x,p1.y), point(p2.x,p1.y));
        draw_line(point(p2.x,p1.y), point(p2.x,p2.y));
        draw_line(point(p2.x,p2.y), point(p1.x,p2.y));
        draw_line(point(p1.x,p2.y), point(p1.x,p1.y));
    }
    void GetBoundingBox(point *ul, point *lr)
    {
        // assume p1 is already the upper left (ul) point
        // and p2 is already the lower right (lr) point
        *ul = p1;
        *lr = p2;
    }
private:
    point p1, p2;
};
```

- 5.1 We would like to define a new shape, triangle, as follows. Please give a **complete implementation** (including the constructor triangle, and the member functions Draw and GetBoundingBox) of the class triangle. [10%]

```
class triangle : public shape {
public:
    triangle(point np1, point np2, point np3);
    void Draw();
    void GetBoundingBox(point *ul, point *lr);
private:
    point p1, p2, p3;
};
```

Note: the three points of a triangle are **not necessarily ordered**; therefore, unlike the rectangle example, your program should be written to discover the correct ul and lr points. The following figure is an example of the bounding box of a triangle.



- 5.2 By using **composite pattern**, we would like to have a new composite shape, picture, which is composed of a number of shapes (including rectangle, triangle, and picture). The class picture is defined as follows. Please give a **complete implementation** of the picture class. [20%]

```
class picture : public shape {
public:
    picture();
    ~picture();
    void Draw();
    void GetBoundingBox(point *ul, point *lr);
    void AddShape(shape *s); // add a new shape into this picture
private:
    vector<shape *> shapes; // the shapes stored in this picture
};
```

Note: The following figure is an example of the bounding box of a picture, which consists of two triangles and one rectangle. Again, your program should be written to discover the correct ul and lr points.

