

程式語言 2022/6/13 (共 12 題)

1. (10%) 底下的程式碼片段當中，

(a) 有兩個 `if` 和一個 `else`，可能存在什麼樣的語意判斷問題？(4%)

(b) 請舉出兩種解決方式？(6%)

```
if (sum == 0)
if (count == 0)
result = 0;
else result = 1;
```

2. (6%) 如果一個程式語言支援了哪兩種控制陳述 (control statement)，便能實作出所有可用流程圖 (flowchart) 表現的演算法？

3. (8%) 如果在一個程式語言裡，打算加入 `if` 表示式 (expression) 來支援底下的程式碼片段，

(a) 需要考慮什麼問題？(4%)

(b) 如何解決？(4%)

```
int x = 5;
int y = x if x > 0;
```

4. (10%) 執行底下的 C++ 程式碼片段，會印出 `a` 與 `b` 的值各為多少？

```
void fun(int x, int &y) {
    x = 10;
    y = 20;
}
int main() {
    int a = 5;
    int b = 10;
    fun(a, b);
    printf("a = %d, b = %d\n", a, b);
    return 0;
}
```

5. (9%) 對於底下以 JavaScript 語法寫成的程式碼片段，分別考慮在不同參照環境 (referencing environment) 下執行時會印出的值：

- (a) Shallow binding
- (b) Deep binding
- (c) Ad hoc binding

```
var x;  
function sub1() {  
    console.log(x); // print the value of x  
};  
function sub2(subx) {  
    var x;  
    x = 20;  
    subx();  
};  
function sub3() {  
    var x;  
    x = 30;  
    sub2(sub1)  
};  
x = 10;  
sub3();
```

6. (10%) 在一個程式語言裡，讓使用者可設計自定義的資料抽象型別 (abstract data type, ADT) 有哪兩個好處？

7. (9%) 關於 C++對於資料抽象型別 (abstract data type, ADT) 的支援，
- (a) 除了 struct 之外，主要是以哪個機制支援？
 - (b) 參數化資料抽象型別 (parameterized ADT) 是以何種機制支援？
 - (c) 參數化資料抽象型別是在編譯時期 (compile time) 或是執行時期 (runtime) 被填入具體型別進行轉換？

8. (10%) 執行底下的 C++ 程式碼，

(a) 將會呼叫哪一個 draw() 而印出何種字串？(5%)

(b) 該機制稱為什麼？(5%)

```
class Shape {  
    public:  
        virtual void draw() = 0;  
};  
class Circle : public Shape {  
    public:  
        void draw() { cout << "Circle::draw()"; }  
};  
class Rectangle : public Shape {  
    public:  
        void draw() { cout << "Rectangle::draw()"; }  
};  
class Square : public Rectangle {  
};  
  
int main() {  
    Shape* s = new Square();  
    s->draw();  
}
```

9. (10%) 任務同步 (task synchronization) 是指控制任務執行順序的機制。當任務之間共用某些資料時，可以考慮有哪兩種類型的同步？

10. (5%) 在下列三個提供任務同步的機制裡，何者是將共用的資料結構與操作包裝為抽象資料型別？(1) Semaphores, (2) Monitors, (3) Message passing

11. (5%) 底下含有例外處理 (exception handling) 的 C++ 程式碼片段會印出何種字串？

```
try {  
    try {  
        throw 10;  
    }  
    catch (int i) {  
        throw 'c';  
    }  
    catch (char) {  
        cout << "inner";  
    }  
}  
catch (char) {  
    cout << "outer";  
}
```

12. (8%) 底下含有例外處理 (exception handling) 的 C++ 程式碼片段含有哪兩個錯誤？

```
try {  
    throw 10;  
}  
catch (...) {  
    cout << "catch all";  
}  
catch (int) {  
    cout << "catch int";  
}  
catch (int i) {  
    cout << "catch int";  
}
```