

國立清華大學 106 學年度碩士班考試入學試題

系所班組別：服務科學研究所 考試科目（代碼）：計算機概論（4802）

共 4 頁，第 1 頁 *請在【答案卷、卡】作答

1. (10%) The following code written in a dynamically typed, object-oriented language, defines a recursive function that takes a nested array as its arr parameter.

```
def fun_print(arr) {
  if (arr.length() > 1) {
    fun_print(arr[1..arr.length-1])
  }

  if arr[0].type_of(Array) {
    fun_print(arr[0])
  } else {
    print(arr[0])
  }
}
```

What would you expect this function to output if called as follows:

```
fun_print( [[['A', 'B', 'C']], 'D', ['E', 'F']] )
```

- (A) A D E B F C (C) A B C D E F (E) F E D C B A
(B) E F D A B C (D) A B C E F D (F) D E F A B C

2 (10%). The following code, written in a statically typed, object-oriented language, uses a class called LinkedList to implement a singly linked list of instances of class Node. Calling .head() on an instance of LinkedList returns the first Node in the list.

```
LinkedList linkedList = new LinkedList();
linkedList.add( new Node("A") );
linkedList.add( new Node("B") );
linkedList.add( new Node("C") );
linkedList.add( new Node("D") );
linkedList.add( new Node("E") );

Node current = linkedList.head();
Node bar = linkedList.head();
int foo = 0;

do {
  foo++;
  if(foo%2 == 0){
    bar = bar.next();
  }
  current = current.next();
} while(current != null)

System.out.println("foo is: " + foo);
System.out.println("bar is: " + bar);
```

What is printed by this code?

- (A) foo is: 4 (C) foo is: 2 (E) foo is: 4
bar is: C bar is: B bar is: E
(B) foo is: 5 (D) foo is: 5 (F) foo is: 2
bar is: E bar is: C bar is: D

國立清華大學 106 學年度碩士班考試入學試題

系所班組別：服務科學研究所 考試科目（代碼）：計算機概論（4802）

共 4 頁，第 2 頁 *請在【答案卷、卡】作答

To answer the next three questions, pick from the following data structures:

- | | |
|---------------|----------------|
| 1. Hash Table | 4. Linked List |
| 2. Array | 5. Binary Tree |
| 3. Stack | 6. Queue |

3. (10%) Which of the above data structures do we usually use if we want to directly retrieve any element by supplying a lookup value:

- | | |
|-----------------|-----------------|
| (A) 2, 3, and 5 | (D) 3, 4, and 5 |
| (B) 4 and 6 | (E) 2, 3, and 5 |
| (C) 1, 2, and 6 | (F) 1 and 2 |

4. (10%) Which of these structures would we use to keep a list of recently used files, such that we can quickly retrieve and remove files in most recently used order?

- | | | |
|-------|-------|-------|
| (A) 1 | (C) 3 | (E) 5 |
| (B) 2 | (D) 4 | (F) 6 |

5. (10%) Which of these data structures would we use to enter a list of tasks, such that we can retrieve and remove the tasks in the same order they were entered?

- | | | |
|-------|-------|-------|
| (A) 1 | (C) 3 | (E) 5 |
| (B) 2 | (D) 4 | (F) 6 |

6. (10%) Which of these data structures would we use to create an *expert system*, that asks 'yes' and 'no' questions to find the answer to a problem. For example:

Do you like coding your own applications (y/n)? "Y"

Do you prefer to work alone by yourself (y/n)? "N"

Do you like to design your own services (y/n)? "Y"

Do you want to be have your own startup (y/n)? "Y"

We recommend the following major: Service Science

- | | | |
|-------|-------|-------|
| (A) 1 | (C) 3 | (E) 5 |
| (B) 2 | (D) 4 | (F) 6 |

國立清華大學 106 學年度碩士班考試入學試題

系所班組別：服務科學研究所 考試科目（代碼）：計算機概論（4802）

共 4 頁，第 3 頁 *請在【答案卷、卡】作答

7. (40%) This problem requires you to improve the structure of code.

You must rewrite the code below as follows:

- Use object-orientated principles to remove repetition in the code
- Your new code must continue working with existing test code (see top of *next page*)
- Follow the OOP syntax of the programming language (see bottom of *next page*)

```
class Student
  var name
  var must_teach

  def new(full_name)
    name = full_name
    must_teach = false
  end

  def notify_enroll(semester)
    puts("#{name} must enroll for classes in {{semester}}")
  end
end

class Lecturer
  var name
  var must_teach
  var must_research

  def new(full_name)
    name = full_name
    must_teach = true
    must_research = false
  end

  def notify_grades(semester)
    puts("#{name} must submit grades for {{semester}}")
  end
end

class Professor
  var name
  var must_teach
  var must_research

  def new(full_name)
    name = full_name
    must_teach = true
    must_research = true
  end

  def notify_grades(semester)
    puts("#{name} must submit grades for {{semester}}")
  end

  def notify_grant(semester)
    puts("#{name} must submit a grant in {{semester}}")
  end
end
```

國立清華大學 106 學年度碩士班考試入學試題

系所班組別：服務科學研究所 考試科目（代碼）：計算機概論（4802）

共 4 頁，第 4 頁 *請在【答案卷、卡】作答

Here is working test code that must still work after your refactoring:

```
# TEST CODE
lee = Student.new('Li-Hsieh Huang')
hsu = Lecturer.new('Powe Hsu')
guo = Professor.new('Peishan Guo')

everyone = [lee, hsu, guo]

for person in everyone
  if (not person.must_teach)
    person.notify_enroll('Spring 2016')
  end

  if (person.must_teach)
    person.notify_grades('Fall 2015')
  end

  if (person.must_teach) && (person.must_research)
    person.notify_grant('Spring 2016')
  end
end

# TEST OUTPUT
# => Li-Hsieh Huang must enroll for classes in Spring 2016
# => Powei Hsu must submit grades for Fall 2015
# => Peishan Guo must submit grades for Fall 2015
# => Peishan Guo must submit a grant in Spring 2016
```

Here is sample OOP code to help you understand the syntax of inheritance:

```
# SAMPLE OOP CODE:
class Automobile
  var brand # var defines a public instance variable
  var mode

  def new(brand_name) # new is a constructor method
    brand = brand_name
    mode = 'ground'
  end
end

class Car < Automobile # Car inherits from Automobile class
  var model
  var num_wheels

  def new(brand_name) # Calls the parent's constructor method
    super(brand_name)
    num_wheels = 4
  end

  def describe()
    puts "The {{brand}} {{model}} has {{num_wheels}} wheels"
  end
end

my_car = Car.new('Luxgen')
my_car.model = 'U6 Turbo'
my_car.describe

# OUTPUT
# => "The Luxgen U6 Turbo has 4 wheels"
```