淡江大學96學年度轉學生招生考試試題

系別：資訊工程學系三年級　　科目：程 式 語 言

Section I: True/False Questions (20 points in total, each question is weighted 2 points)

1. One of the advantages of interpreted languages is their faster execution time compared to compiled languages.

2. If the grammar for a language is ambiguous, then some valid program in that language has more than one parse tree.

3. BNF grammars do not allow left-recursive rules.

4. Even though Java uses a garbage collector, there can still be memory leaks due to circular data structures that appear to be referenced even though they are not.

5. Dynamic type checking adds some execution-time overhead, but improves reliability of programs.

6. In an attribute grammar, a node's synthesized atttributes are based on the values associated with that node's ancestors.

7. Axiomatic semantics provide a good way of documenting the design of a program.

8. One advantage the garbage collection technique over a reference counter scheme for automatic memory management is that it can handle negative reference counts.

9. A programming language is considered to be strongly typed if every variable must be associated with a single primitive data type.

10. A union type is a type that can store different type values at different times during the execution of a program.

Section II: Multiple Choice Questions (as indicated by each question, there may be more than one correct answers to the question. 10 points in total, each question is weighted 2 points)

1. Which of these are **not** legal identifiers in C programming language. Select the **six** correct answers.

A. number_1	B. for8_
C. $1234	D. enum
E. 4alpha	F. _abcd
G. xy+abc	H. this
I. account-num	J. very_long_name

2. Which of the following are keywords in C++. Select the **two** correct answers.

A. friend	B. NULL
C. implement	D. synchronized
E. throw	

3. Which of the following are Java keywords. Select the **four** correct answers.

A. friend	B. strictfp
C. volatile	D. protect
E. instanceof	F. finally

4. Which of these are C keywords. Select the **three** correct answers

A. TRUE	B. volatile
C. transient	D. native
E. interface	F. then
G. continue	

5. The Java statement

```
public class LinkedListStack {}
```

creates: (only one correct answer)

A. new class	B. new object
C. new reference variable	
D. new container to hold objects	

本試題雙面印製

Section III: Programming examples (30 points in total, each question is weighted 6 points)

1. Consider the following C++-style function and a suitable call to the function `foo()`. What is the final value of `n` if parameters to the function `foo()` are passed by

(1) value-result
(2) reference

```
int n = 5;  // global
void foo(int x, int y){ ++x; ++y;}
...
foo(n, n);   // the call
```

2. Consider that `Parent` and `Child` classes are defined in two different files as below:

```
class Parent{
  public Parent(){
  System.out.println("Parent");
  }
}

class Child extends Parent{
  public Child(int x){
  System.out.println("Child");
}

public static void main(String [] args){
  Child c=new Child(10);
  }
}
```

What will be the output if you try to compile and run above program? (one correct answer)

A. It will not compile.
B. It will compile successfully and print "Parent" and then "Child."
C. It will compile successfully and print "Child" and then "Parent."
D. It will compile successfully, but not run.

3. Consider following code:

```
public class OuterClass{
  class InnerClass{ }
    public void innerClassDemo(){
    //Explicit instance of InnerClass
    }
}
```

How can you explicitly create an instance of `InnerClass`? (two correct answers)

A. InnerClass i=InnerClass();
B. InnerClass i=OuterClass.InnerClass();
C. InnerClass i=new OuterClass ().new InnerClass();
D. OuterClass.InnerClass i=new OuterClass.InnerClass();

4. Consider the following code:

```
/** File Thread1.java */
class Thread1 implements Runnable{
  public void run(){
    System.out.println("Running Thread1");
    }
} /** End of file Thread1.java */

/** Thread2.java */
class Thread2 extends Thread{
  public void run(){
    System.out.println("Running Thread2");
    }

  public static void main(String [] args){
    Thread1 t1= new Thread1();
    Thread t2=new Thread2(t1);
    t1.start();
    t2.start();
    }
} /** End of Thread2.java*/
```

If you try to compile and run above code what will be result? (one correct answer)

A. "Running thread1" following "Running thread2"
B. "Running thread2" following "Running thread1"
C. It will not compile because in `Thread1` and `Thread2` `start()` is not defined.
D. It will not compile because constructor invoked to create `Thread2` with arguments (`Thread1`) is not defined.

5. Consider that class Employee and Salesman are in different file called Employee.java and Salesman.java:

```
/** Employee.java file*/
public class Employee{
int salary=1000;
public int getSalary(){return salary;}
} /** End of Employee.java file*/
/** Salesman.java file*/
public class Salesman extends Employee{
int commission =100;
public int getSalary(){
        return salary+commission;}
public static void main(String [] args){
  Salesman sm = new Salesman();
  Employee em = sm;
  System.out.println(em.getSalary());
  }
} /** End of Salesman.java file*/
```

What will be result if you try to compile and run above code? (one correct answer)

A. Compiler error reported , "Type mismatch: Cannot convert from Salesman to Employee."
B. It compile successfully and outputs 1000.
C. It compiles successfully and outputs 1100.
D. None of the above.

6. How can you declare a overloaded method? (two correct answers)

A. Reusing the method name with different arguments and same return type.
B. Reusing the method name with different arguments and different return type.
C. Reusing the name with identical arguments and return type.
D. None of the above.

7. When can't you override a method?
(one correct answer)
A. When method is declared abstract.
B. When method is declared final.
C. When method is declared private.
D. When method is declared static.

8. What is the Java expression that can be used to represent number of elements in an array named arrayOne ?

9. The Java statement
```
Object element = new Object();
```
creates: (two correct answers)
A. new class
B. new object;
C. new reference variable;
D. new container to hold objects

10. Consider the following Java code:

```
interface A {
  void m(A y);
}

interface C extends A {
  A n(C y);
}

class B implements A {
  int x;
  void m(A y){
    return;
  }
}

class D extends B {
  int y;
}

class E extends B implements C {
  A n(C z){
    return z;
  }
}
```

(1) Name the types, classes and interfaces that have been created.

	Name
Types	
Classes	
Interfaces	

(2) Which of the following is true ? (two correct answers)
(i) A inherits from B
(ii) A inherits from C
(iii) B inherits from A
(iv) D inherits from C
(v) D inherits from B
(vi) E inherits from C
(vii) E inherits from B
(viii) E inherits from A

本試題雙面印製

Section IV: Programming examples (40 points in total)

Problem 1: (20 points) Parameter passing

Consider the following program written in C syntax: For each of the following parameter-passing methods, what is the output of the program:

```
void main() {
      int value = 2, list[5] = {1,3,4,7,9};
      swap(value,list[0]);
      printf("%d and [%d,%d,%d,%d,%d]\n",
               value,list[0],list[1],list[2],list[3],list[4]);

      swap(list[0],list[1]);
      printf("%d and [%d,%d,%d,%d,%d]\n",
               value,list[0],list[1],list[2],list[3],list[4]);

      swap(value,list[value]);
      printf("%d and [%d,%d,%d,%d,%d]\n",
               value,list[0],list[1],list[2],list[3],list[4]);
   return 0;
   }
   void swap(int x, int y) {
      int temp;

      temp = x;
      x = y;
      y = temp;
   }
```

(1). Passed by value.
(2). Passed by reference.
(3). Passed by name.
(4). Passed by value-result.

Problem 2: (10 points) Scoping

Consider the following program written in C-like syntax:

```
int x;
void setX(int n) {
   x = n;
}
void printX() {
  printf("%d\n",x);
}
void first() {
  setX(1);
  printX();
}
void second() {
  int x;
  setX(2);
  printX();
}
int main() {
  setX(0);
  first();
  printX();
  second();
  printX();
}
```

(1). What does this program print if the language uses **static scoping**?

(2). What does this program print if the language uses **dynamic scoping**?

淡江大學96學年度轉學生招生考試試題

系別：資訊工程學系三年級　　科目：程 式 語 言

Problem 3: (10 points) Functions as Parameters
Consider the following program written in C-like syntax:

```
int x;
void setX(int n) {
    x = n;
}
void printX() {
   printf("%d\n",x);
}
void foo(void(S*)(int m),
         void(P*)(void), int n) {
   int x = 0;
   if ((n == 1) || (n == 3))
      setX(n);
   else
      S(n);
   if ((n == 1) || (n == 2))
      printX;
   else
      P();
}
int main() {
   setX(0); foo(setX,printX,1); printX();
   setX(0); foo(setX,printX,2); printX();
   setX(0); foo(setX,printX,3); printX();
   setX(0); foo(setx,printX,4); printX();
}
```

Assume that the language uses **dynamic scoping**.
(1). What does the program print if the language uses **shallow binding**?

(2). What does the program print if the language uses **deep binding**?