94學年度第2學期物件導向程式設計期中測驗

姓名		系級		學號	

(True/False) (30%)

1. (Ch1) A variable of type `boolean` can be explicitly converted to that of type `int`.
2. (Ch1) An advantage of using the Unicode character set is that it easily handles languages other than English.
3. (Ch1) Applets were designed to run as stand-alone applications.
4. (Ch1) In Java, allocated memory that is no longer needed should be deallocated.
5. (Ch2) Every Java program automatically imports the `java.util` package.
6. (Ch3) The equality operator (==) may be used to test if two string objects contain the same value.
7. (Ch3) The three expressions at the start of a `for` statement are separated by two commas.
8. (Ch3) In a `switch` statement, the `default` case is always executed.
9. (Ch4) An object of class A is an instance of class A.
10. (Ch4) The Java language supports global variables.
11. (Ch4) Java supports operator overloading.
12. (Ch4) Only the default constructor has the `this` parameter.
13. (Ch4) Because of pass-by-value, the passed contents of the object can be changed in the called method, but the original object reference is never changed.
14. (Ch5) In a static method, you may use the `this` parameter either explicitly or implicitly.
15. (Ch5) The `String` class is a mutable class.

(Choices) (30%)

1. (Ch1) Identify the invalid Java identifier.
   (a) 1Week (b) Week1 (c) amountDue (d) amount_due
2. (Ch1) In Java, source code is compiled into object code called _____.
   (a) Bit-code (b) Class code (c) Method code (d) Byte-code
3. (Ch1) The value of the expression `(int)27.6` evaluates to:
   (a) 28 (b) 27 (c) 26 (d) None of the above.
4. (Ch1) What is the value of the variable c in the statements that follow?
   ```
   String phrase = "Make hay while the sun is shining.";
   char c = phrase.charAt(10);
   ```
   (a) w (b) h (c) i (d) None of the above
5. (Ch1) To mark a block comment for inclusion in the Javadoc documentation, the block must be delimited by:
   (a) /** */ (b) */* */ (c) **/ /* (d) **/ */

94學年度第2學期物件導向程式設計期中測驗

6. (Ch3) The controlling expression for a `switch` statement includes all of the following types except:
   (a) char (b) int (c) byte (d) double
7. The looping mechanism that always executes at least once is the _____ statement.
   (a) `if...else` (b) `do...while` (c) `while` (d) `for`
8. 14) A _____ statement terminates the current iteration of a loop.
   (a) Break (b) Continue (c) Switch (d) Assert
9. (Ch3) When using a compound Boolean expression joined by an `&&`(AND) in an `if` statement:
   (a) Both expressions must evaluate to true for the statement to execute.
   (b) The first expression must evaluate to true and the second expression must evaluate to false for the statement to execute.
   (c) The first expression must evaluate to false and the second expression must evaluate to true for the statement to execute.
   (d) Both expressions must evaluate to false for the statement to execute.
10. (Ch4) Java has a way of officially hiding details of a class definition. To hide details, you mark them as _____.
    (a) `public` (b) `protected` (c) `private` (d) all of the above
11. (Ch4) 17) The name of a method and the list of _____ types in the heading of the method definition is called the method signature.
    (a) parameter (b) argument (c) return (d) primitive
12. (Ch5) Only _____ copy/copies of a static variable are available to objects of a class.
    (a) one (b) two (c) three (d) none of the above
13. (Ch5) All of the following are wrapper classes except:
    (a) `String` (b) `Integer` (c) `Character` (d) `Double`
14. (Ch5) When you use the assignment operator with variables of a class type, you are assigning a:
    (a) value (b) primitive type (c) local variable (d) reference
15. (Ch5) A condition that allows a programmer to circumvent the private modifier and change the private instance variable is called:
    (a) a copy constructor (b) a privacy leak (c) a class invariant (d) an anonymous object

Short Answer: (40%)

1. List the primitive data types Java supports. Indicate the number of bytes each type used.

   boolean 1
   byte 1
   short 2
   int 4
   char 2
   long 8
   float 4
   double 8

2. Describe the detailed construction process of the following statements:
   ```
   Integer tmp = new Integer(19);
   ```

3. Correct the following code:
   ```
   public class EX {
       public void main(String args) {
           long x = 0;
           for (x = 1; x < 20; x++) {
               switch (x % 2) {
                   case 1: System.out.println(x + " is odd");
                   case 0: System.out.println(x + " is even");
               }
           }
       }
   }
   ```

(C) Zero
(D) The value cannot be predicted

3. (Ch7) Which statements are true?
   a. Overriding is just another term for overloading
   b. Overloading only can be done within an inheritance structure
   c. Overriding only can be done within an inheritance structure
   d. You can overload a constructor
   e. You can override a constructor
(A) Only statements (c) and (d) are true
(B) All the statements are true
(C) Only statement (d) is true
(D) True statements: (b), (c), (d), and (e)
(E) None of the above

4. (Ch7) If the `final` modifier is added to the definition of a method, this means:
(A) The method may be redefined in the derived class.
(B) The method may be redefined in the sub class.
(C) The method may not be redefined in the derived class.
(D) None of the above.

5. (Ch7) A method or instance variable modified by `protected`:
(A) can not be accessed by name inside its own class definitions.
(B) can not be accessed by name inside any class derived from it.
(C) can not be accessed by name in the definition of any class in the same package.
(D) can be accessed by name in any other class (that is, other than classes named in (A)-(C).).

6. (Ch8) Which statement is true?
(A) You can declare a class to be both `final` and `abstract` so long as it doesn't contain any methods
(B) You cannot declare a class to be both `final` and `abstract`
(C) There is no problem declaring a class to be both `final` and `abstract`
(D) You can declare a class to be both `final` and `abstract` so long as it also is declared to be `static`

7. (Ch8) ____ refers to the ability to associate many meanings to one method name by means of the late binding mechanism.
(A) Inheritance (B) Encapsulation (C) Polymorphism (D) None of the above

8. (Ch8) An abstract method cannot be modified by:
(A) `public` (B) `protected` (C) `private` (D) none of the above

9. (Ch9) The catch block has _____ parameters.
(A) zero (B) one (C) two (D) three

10. (Ch9) If a method throws an exception, and the exception is not caught inside the method, then the method invocation:
(A) terminates (B) transfers control to the catch block (C) transfers control to the exception handler
(D) none of the above

11. (Ch9) `ArrayIndexOutOfBoundsException` is a descendent of the class `RuntimeException`. This means:
(A) the exception must be caught (B) a finally block must be included
(C) the exception does not have to be explicitly caught (D) none of the above

12. A (Ch13) An interface and all of its method headings are normally declared to be:
(A) `public` (B) `private` (C) `protected` (D) package access

C (Ch13) An interface may contain:
(A) instance variables (B) primitive variables (C) constant variables (D) all of the above

A (Ch13) Inner classes available outside the scope of their outer class are modified by the keyword:
(A) `public` (B) `private` (C) `protected` (D) Package access

姓 名		系 級		學 號	

(True/False) (30%)
1. (Ch6) An arrays length instance variables value can be changed by a program
2. (Ch6) An array of `chars` is the same as a `String` in Java.
3. (Ch6) A variable of an enumerated type can have the special value `null`.
4. (Ch7) A derived class contains only `public` instance variables and `public` methods from the base class.
5. (Ch7) Overriding is when a derived class redefines a method from the base class.
6. (Ch7) `Private` methods of the base class are not available for use by derived classes.
7. (Ch8) Java allows an instance of an `abstract` class to be instantiated.
8. (Ch8) Java uses late binding with `private` methods, methods marked `final`, or `static` methods.
9. (Ch8) An `abstract` method serves as a placeholder for a method that must be defined in all derived classes.
10. (Ch9) When an exception is thrown, the code in the surrounding `try` block continues executing and then the `catch` block begins execution.
11. (Ch9) Exceptions that must follow the Catch or Declare Rule are often called checked exceptions.
12. (Ch9) You can not place a `try` block and its following `catch` blocks inside a larger `try` block or inside a larger `catch` block.
13. (Ch13) A class may only implement one interface.
14. (Ch13) You can not derive an interface from a base interface.
15. (Ch13) Java source code that contains a class with an inner class, when compiled, will produce a separate .class file for the inner class.

(Choices - Single) (30%)
1. (Ch6) A method to compute the sum of all elements in an array of `int` is needed. The following proposed method is incomplete:

```
1.  public int total( int[] x )
2.  {
3.     int i, t = 0 ;
4.     <- select statement to go here
5.     {
6.        t += x[ i++ ] ;
7.     }
8.     return t ;
9.  }
```

What is the correct statement for line 4?
(A) `for ( int i = 0; i < x.length; )`
(B) `for ( i = 0; i < x.length; )`
(C) `for ( i = 0; i < x.length; i++ )`
(D) `for ( i = 1; i <= x.length; i++ )`
(E) None of the above

2. (Ch6) After the following code has been executed, what will the first element of the array contain?

```
String[] types = new String[ 20 ];
```

(A) An empty string
(B) The null value

```
public class B extends A {
    public int b;
    public void m1() {
        System.out.println(20);
    }
}
```

Given the above two class declarations and

```
A[] ref = new A[2];
ref[0] = new A();
ref[1] = new B();
```

(1) (true or false) ref[1].b = 20; is legal.
(2) (true or false) ref[1].m1(); This statement will print out "20"
(3) (true or false) class B inherits the constructor from class A
(4) System.out.println(ref[1].ino() + ref[2].ino());
(true or false) The above statement will print out "2"

4. Read the following class definition:

```
1. public class E1 extends Exception {
2. }
3. Public class E2 extends E1 {
4. }
5. Public class E3 extends Exception {
6. }
7. public class A {
8.     public void m1() throws E1 {
9. }
10. public class B extends A {
11.     protected void m1() throws E2, E3 {
12. }
13. public class Test {
14.     public static void main(String[] args) {
15.         try {
16.             A ref = new B();
17.             ref.m1();
18.         } catch (E2 e) {
19.             e.printStackTrace();
20.         }
21.     }
22. }
```

Will these codes be compiled without error? If any error occurred, describe your answer to correct the codes without change the definitions of class E1, E2, E3, B.

5. Read the following class definition:

```
public class A {
    public static final int SUNDAY    = 0;
    public static final int MONDAY    = 1;
    public static final int TUESDAY   = 2;
    public static final int WEDNESDAY = 3;
    public static final int THURSDAY  = 4;
    public static final int FRIDAY    = 5;
    public static final int SATURDAY  = 6;
}
public class Test {
    public static void print(int day) {
        switch (day) {
        case A.SUNDAY: case A.MONDAY:
            System.out.println("Weekend"); break;
        Default:
            Syste.out.println("Weekday"); break;
        }
    }
}
```

Please rewrite the above codes with enum instead of the original constant definitions.

(Ch13) Consider the following outline of the declaration of a normal class with an inner class.

```
1.  public class NormClass {
2.      public class NestedClass {
3.          // methods and variables of NestedClass
4.      }
5.      // methods and variables of NormClass
6.  }
```

Which of the following is the correct way for a method inside NestedClass to refer to the enclosing instance of NormClass?
(A) this
(B) NormClass.this
(C) this.NormClass
(D) this.this
(E) this.super

(Choices - Multiple) (40%)
Suppose you have two classes defined as follows:

```
class ApBase extends Object implements Runnable {}
class ApDerived extends ApBase implements Observer {}
```

Also suppose you have two variables created as follows:

```
ApBase aBase = new ApBase() ;
ApDerived aDer = new ApDerived() ;
```

Which of the following Java statements will compile and execute without error? [Check all correct answers.]
(A) Runnable rn = aDer;
(B) Runnable rn2 = (Runnable) aBase;
(C) Observer ob = aBase;
(D) Observer ob2 = (Observer) aBase;

2. Look at the following class definition:

```
1.  public class DerivedDemo extends Demo {
2.      int M, N, L ;
3.      public DerivedDemo( int x, int y ) {
4.          M = x ; N = y ;
5.      }
6.      public DerivedDemo( int x ) {
7.          super( x );
8.      }
9.  }
```

Which of the following constructor signatures must exist in the Demo class for DerivedDemo to compile correctly? [Check all correct answers.]
(A) public Demo(int a, int b)
(B) public Demo(int c)
(C) public Demo( )
(D) There is no requirement for a constructor in Demo.

3. Read the following class definition:

```
public class A {
    public int a1;
    public static int a2 = 0;

    public void m1() {
        System.out.println(10);
    }
    public int ino() {
        a2++;
        return a2;
    }
    public static void m2() {
    }
    public final void m3() {
    }
```

4. Correct the following constructor without changing the logic.

```
public class Order{
    private String acct = "AAA";
    private String item = "pen";
    private double price = 25.0;
    private int qty = 1;
    public Order (String acct, String item, double price, int qty) {
      this.acct = acct;
      this.item = item;
        price = (price < 0) ? 0 : price;
        qty = (qty < 0) ? 0 : qty;
    }
}
```

if (price < 0) price = 0 else price = price

5. In order to avoid the privacy leakage of class A, please correct the following statement. (hint: rewrite the accessor and mutator of attibutes math with clone constructor of class Credit)

```
public class Student {
    private Credit math = new Credit();
    public Credit getCredit () {
        return math;
    }
    public void setCredit (Credit tmp) {
        math = tmp;
    }
}

public class Credit {
    private int score;
    public Credit (Credit tmp) {
        score = tmp.score;
    }
}
```

return new Credit(math);

math = new Credit(tmp);

base ← A ref = new B(); → derive

B.m();