

JAVA Programming Language Homework II

Student ID: Name:

1. Given the following Java code:

```
1.      class A {
2.          public static void main (String[] args) {
3.              Byte tmp1 = new Byte("1");
4.              Byte tmp2 = new Byte("1");
5.              if (tmp1 == tmp2) System.out.println("true,");
6.              else System.out.println("false,");
7.              if (tmp1.equals(tmp2)) System.out.println("true,");
8.              else System.out.println("false,");
9.          }
10.     }
```

What is the result?

- A. false, false
- B. false, true
- C. true, false
- D. true, true
- E. None of the above

Answer: B

`tmp1 == tmp2` 敘述乃比較兩個 Byte 物件的 Reference value，而 `tmp1.equals(tmp2)` 則是比較兩個物件的內容，其中 `tmp1` 與 `tmp2` 的內容皆為 1。

2. Given the following Java code:

```
1.      class A {
2.          final String s1 = "A.s1";
3.          final String S2 = "Aa.s1";
4.          class B {
5.              String s1;
6.              void m() {System.out.println(???);}
7.              void n() {System.out.println("B");}
8.          }
9.          public static void main(String args[]) {
10.             A g = new A();
11.             g.new B().m();
```

```
12.      }
13.      }
```

What field access expression could be used in place of ??? above to cause the program to print “A.s1” ?

- A. s1
- B. A.s1
- C. ((A)this).s1
- D. A.this.s1
- E. None of the above

Answer: D

因為類別 A 和類別 B 都有變數 s1，所以在內部類別 B 中呼叫外圍類別 A 的成員時，需使用此方法：外圍類別名稱.this. 欲存取資源的名稱。

3. Given the following Java code:

```
1.      Integer i = new Integer(42);
2.      Long l = new Long(42);
3.      Double d = new Double(42.0);
```

Which two expressions evaluate to True?

- A. (i== 1)
- B. (i == d)
- C. (i.equals(42))
- D. (i.equals(d))
- E. (d.equals(i))

Answer: Depend on the version of JDK. C is right in JDK1.5

A、B：比對物件須利用 equals() 方法，使用等號是比對參考位置而非物件內容。

D、E：equals() 提供的是 Integer 的 Object 內容比對。

C：在 JDK1.4 版與 JDK1.4 版以前，equals() 宣告方法為：public Boolean equals(Object obj)，所以 C 的用法不對。但是在 JDK1.5 版以後，加入的 Auto Boxing 功能可進行隱含轉換，所以 C 的作法是可行的。

4. Given the following Java code:

```
11.    class A {
12.        public static final int a = 1;
13.        protected static int b=2;
14.        int c=3;
15.        static class B {
16.            int d=a;
17.            int e=b;
18.            int f=c;
19.        }
20.    class C {
21.        int g=a;
22.        int h=b;
23.        int i=c;
24.    }
25. }
```

What is the result?

- F. Compilation Error at line 10.
- G. Compilation Error at line 8.
- H. Compilation Error at line 2.
- I. Run without any problem.
- J. None of the above.

Answer: B

在 Static Nested Class 中，直接存取外圍類別的成員是不被允許的。

5. Given the following Java code:

```
1.    class B {
2.        private int x = 2;
3.        static A a1 = new A(2,1) {
4.            public A(int tmp) {x(tmp); y(tmp);};
5.            public int m() {return x()+y();}
6.        };
7.        public static void main(String[] args) {
8.            System.out.print(a1.m());
9.        }
```

```
10.     }
11.     abstract class A {
12.         private int x = 4;
13.         private int y = 2;
14.         private int z = 6;
15.         public int x() {return x;}
16.         public void x(int x) {this.x = x;}
17.         public int y() {return y;}
18.         public void y(int y) {this.y = y;}
19.         public abstract int m();
20.     }
```

What is the result?

- A. Prints: 8
- B. Prints: 3122
- C. Compilation fails
- D. Run-time error
- E. None of the above

Answer: C

匿名內部類別是不允許擁有建構子的，所以第 4 行會發生編譯錯誤。

6. Given the following Java code:

```
1.     public class Foo {
2.         Foo() {System.out.println("foo");}
3.         class Bar {
4.             Bar() {System.out.println("bar");}
5.             public void go() {System.out.println("hi");}
6.         }
7.         public static void main (String[] args) {
8.             Foo f = new Foo();
9.             f.makeBar();
10.        }
11.        void makeBar() {
12.            (new Bar() {}).go();
13.        }
14.    }
```

What is the result?

- A. Run-time error.
- B. Compilation fails.
- C. foobarhi
- D. barhi
- E. hi

Answer: C

第一個 Foo 的實例會被建立，因此 Foo 建構子會執行並印出「foo」。

接下來，makeBar()方法會被啟動，他會產生一個 Bar，並執行 Bar 建構子與印出「bar」。

最後，程式會在新的 Bar 實例上啟動 go()方法，並印出「hi」。

7. Given the following Java code:

```
1.    public class HorseTest {
2.        public static void main (String[] args) {
3.            class Horse {
4.                public String name;
5.                public Horse(String s) {
6.                    name = s;
7.                }
8.            }
9.            Object obj = new Horse("Zippo");
10.           Horse h = (Horse) obj;
11.           System.out.println(h.name);
12.       }
13.   }
```

What is the result?

- A. Compilation Error at line 3
- B. Compilation Error at line 9
- C. Compilation Error at line 10
- D. Compilation Error at line 11
- E. Zippo

Answer: E

所有內容完全合法。

8. Given the following Java code:

```
1.    public class HorseTest {
2.        public static void main (String[] args) {
3.            class Horse {
4.                public String name;
5.                public Horse(String s) {
6.                    name = s;
7.                }
8.            }
9.            Object obj = new Horse("Zippo");
10.           System.out.println(obj.name);
11.        }
12.    }
```

What is the result?

- A. Compilation Error at line 3
- B. Compilation Error at line 9
- C. Compilation Error at line 10
- D. Compilation Error at line 11
- E. Zippo

Answer: C

因為使用型態為 Object 的參考變數，只可以讀取在類別 Object 中所定義的組件。

9. Given the following Java code:

```
1.    public abstract class AbstractTest {
2.        public int getNum() {
3.            return 45;
4.        }
5.        public abstract class Bar {
6.            public int getNum() {
7.                return 38;
```

```

8.         }
9.         }
10.        public static void main (String[] args) {
11.            AbstractTest t = new AbstractTest() {
12.                public int getNum() {
13.                    return 22;
14.                }
15.            };
16.            AbstractTest.Bar f = t.new Bar() {
17.                public int getNum() {
18.                    return 57;
19.                }
20.            };
21.            System.out.println(f.getNum() + " " + t.getNum());
22.        }
23.    }

```

What is the result?

- A. 57 22
- B. 45 38
- C. 45 57
- D. Compilation fails
- E. None of the above

Answer: A

此程式可以順利執行，由於抽象類別無法進行實體化，所以此題目先為之建立了非抽象的匿名子類別，然後再 override getNum() 方法，因此，f.getNum() 與 t.getNum() 各自回傳了 57 和 22。

10. Given the following Java code:

```

1.    public class TestObj {
2.        public static void main (String [] args) {
3.            Object o = new Object() {
4.                public boolean equals(Object obj) {
5.                    return true;
6.                }
7.            }

```

```
8.         System.out.println(o.equals("Fred"));
9.         }
10.        }
```

What is the result?

- A. true
- B. Compilation fails because of an error on line 3.
- C. Compilation fails because of an error on line 4.
- D. Compilation fails because of an error on line 8.
- E. Compilation fails because of an error on a line other than 3, 4, or 8.

Answer: E

因為第 3 行是一個陳述式，直到第 7 行才結束，所以在第 7 行處需以分號結束！如果在第 7 行程式後端有加上分號，則答案 A 便會正確，也就是印出由 Object 的匿名子類別所 override 的 equals() 方法所回傳的內容。

11. Given the following Java code:

```
1.         class Foo {
2.             class Bar{ }
3.         }
4.         class Test {
5.             public static void main (String [] args) {
6.                 Foo f = new Foo();
7.                 // Insert code here
8.             }
9.         }
```

Which statement, inserted at line 7, creates an instance of Bar?

- A. Foo.Bar b= new Foo.Bar();
- B. Foo.Bar b = f.new Bar();
- C. Bar b = new f.Bar();
- D. Bar b = f.new Bar();
- E. Foo.Bar b = new f.Bar();

Answer: 只有 B 的語法是對的，參考宣告中，需同時使用外部與內部類別的名稱，然後使用對象為外部類別的參考來 new 內部類別。