Midterm Examination Problem Sheet

TIME: 04/21/2009, 14:20–17:20

This is a closed-book exam. Any form of cheating or lying will not be tolerated. Students can get zero scores and/or fail the class and/or be kicked out of school and/or receive other punishments for those kinds of misconducts.
Both English and Traditional Chinese (if suited) are allowed for answering the questions. We do not accept any other languages.

1 Let's Go Students (20%)

```
1  class Student{
2    static final int CSIE = 9020;
3    static final int IM = 7050;
4    int dept;  int ID;  String name;
5    public Student(int ID){ this.ID = ID; }
6    public static Student new_CSIE_Student(int ID){
7      Student s = new Student(ID);
8      s.dept = CSIE;
9      return s;
10   }
11   public void show(){
12     System.out.print(dept); System.out.print(", ");
13     System.out.print(ID); System.out.print(", ");
14     System.out.println(name);
15   }
16 }
17 public class StudentDemo{
18   static Student s1;
19   Student s2;
20   public static void main(String[] argv){
21     /* CODE */
22   }
23 }
```

Please write down the output when the `/* CODE */` part is replaced by the following lines (respectively for each subproblem). If you think there is a compile error, write "compile error" or "hahaha." If you think there is a run-time error (exception), write down "run-time error" or "ohohoh."

(1) `System.out.println(Student.CSIE);`

(2) `System.out.println(Student.ID);`

(3) `System.out.println((new Student()).name);`

(4) `System.out.println((new Student(54)).dept);`

(5) `Student s;  s.show();`

(6) `Student s = new Student(54);  s.show();`

(7) `s1.show();`

(8) `s2.show();`

(9) `Student s = Student.new_CSIE_Student(62); s.name = "Luc";  s.show();`

(10) `Student s = null; System.out.println(s.IM);`

2 Object Lifecycle (20%)

Please answer the questions under //. You can assume that there is no compile nor run-time errors, and the JVM runs normally.

```
1  class Person{
2    static int count = 0;
3    String name;
4    Person best_friend;
5    public Person(String name){ this.name = name; count++; }
6    protected void finalize() throws Throwable { count--; }
7  }
8  public class PersonDemo{
9    public static void main(String[] argv){
10     Person.count = 0;
11     for(int i = 0; i < 100; i++){
12       Person p = new Person("" + i);
13     }
14     //(1): the maximum possible value of count
15     //(2): the minimum possible value of count
16     //Next, assume that Person.count is at its minimum possible value
17     Person.count = 0;
18     Person q = null;
19     for(int i = 0; i < 100; i++){
20       Person p = new Person("" + i);
21       q = p;
22     }
23     //(3): the maximum possible value of count
24     //(4): the minimum possible value of count
25     //Next, assume that Person.count is at its minimum possible value
26     Person.count = 0;
27     Person r = null;
28     for(int i = 0; i < 100; i++){
29       Person p = new Person("" + i);
30       p.best_friend = r;
31       r = p;
32     }
33     //(5): the maximum possible value of count
34     //(6): the minimum possible value of count
35     //Next, assume that Person.count is at its minimum possible value
36     Person.count = 0;
37     Person george = new Person("George");
38     george.best_friend = new Person("Mary");
39     george.best_friend.best_friend = george;
40     //(7): the maximum possible value of count
41     //(8): the minimum possible value of count
42     //Next, assume that Person.count is at its minimum possible value
43     //NOTE: Person.count is not reset to 0!!!
44     Person mary = george.best_friend;
45     Person bob = new Person("Bob");
46     bob.best_friend = new Person("Mary");
47     mary.best_friend = george;
48     bob.best_friend = george.best_friend;
49     //(9): the maximum possible value of count
50     //(10): the minimum possible value of count
51   }
52 }
```

3 Method Invocation (20%)

Assume that `OOPUtil.is_prime(n)` returns whether the integer `n` is a prime (as a boolean `true` or `false`). Please answer the questions under `//`. You can assume that there is no compile nor run-time errors, and the JVM runs normally.

```
1  class CrazyPrime{
2    static final int MAXN = 20;
3    static final CrazyPrime[] result = new CrazyPrime[MAXN];
4
5    int value;
6
7    static CrazyPrime compute(int n, int level){
8      //Let's assume that 0 <= n < MAXN
9      level++;
10     System.out.print("l"); System.out.print(level); System.out.print("␣");
11     System.out.print("n"); System.out.print(n); System.out.print("␣");
12
13     int res = 0;
14     if (result[n] == null){
15       if (n <= 2){
16         res = 1;
17       }
18       else if (OOPUtil.is_prime(n)){
19         res = CrazyPrime.compute(n-1, level).value;
20         res += CrazyPrime.compute((n-1)/2, level).value;
21       }
22       else{
23         res = CrazyPrime.compute(n-1, level).value;
24       }
25
26       result[n] = new CrazyPrime();
27       result[n].value = res;
28     }
29     System.out.print("r"); System.out.print(result[n].value);
30     System.out.print("␣");
31     return result[n];
32   }
33 }
34 public class CrazyPrimeDemo{
35   public static void main(String[] argv){
36     CrazyPrime c2 = CrazyPrime.compute(2, 0);
37     System.out.println(""); //(1): Write down the output before the new line
38     CrazyPrime c3 = CrazyPrime.compute(3, 0);
39     System.out.println(""); //(2): Write down the output before the new line
40     CrazyPrime c6 = CrazyPrime.compute(6, 0);
41     System.out.println(""); //(3): Write down the output before the new line
42     CrazyPrime c7 = CrazyPrime.compute(7, 0);
43     System.out.println(""); //(4): Write down the output before the new line
44     c7.compute(5, 0);
45     System.out.println(""); //(5): Write down the output before the new line
46   }
47 }
```

4 Encapsulation (20%)

```
1  /* SpokesPerson.java */
2  package prototype;
3  public class SpokesPerson{
4    private String story;
5
6    public void tell_public_story(){ }
7    void tell_default_story(){ }
8    protected void tell_protected_story(){ }
9    private void tell_private_story(){ }
10
11   private void talk_to_myself(){
12     /* CODE1 */
13   }
14 }
```

When the `/* CODE1 */` part is replaced by the following lines (respectively for each subproblem), if you think there is a compile error, please write "compile error" or "hahaha." Otherwise write "safe."

(1) `story = "POO is the best BBS.";`

(2) `tell_private_story();`

(3) `tell_public_story();`

(4) `tell_protected_story();`

(5) `tell_default_story();`

```
1  /* HeadSpokesPerson.java */
2  package extensions;
3  class HeadSpokesPerson extends prototype.SpokesPerson{
4    public void talk_to_media(){
5      /* CODE2 */
6    }
7  }
```

Now, consider another java source file above. When the `/* CODE2 */` part is replaced by the following lines (respectively for each subproblem), if you think there is a compile error, please write "compile error" or "hahaha." Otherwise write "safe."

(6) `story = "POO is the best BBS.";`

(7) `talk_to_myself();`

(8) `tell_public_story();`

(9) `tell_protected_story();`

(10) `tell_default_story();`

5 Inheritance (20%)

```
1  class BBSBoardSystem { };
2  class BBS{ public BBSBoardSystem board_system; int ip; }
3  class Casino{ private int n_games; }
4  class PTTCasino extends Casino { };
5  class POOCasino extends PTTCasino { };
6  class PTTBBS extends BBS{ public PTTCasino casino; }
7  class POOBBS extends BBS{ private POOCasino casino; }
8  class NetBBS extends BBS{ PTTBBS ptt; POOBBS poo; }
9  class PPPBBS extends POOBBS{ }
```

If you know that in Java, an Object-type instance occupies B bytes of space for its instance variables, an integer-type variable occupies 4 bytes, and a reference-type variable occupies R bytes. How many bytes does an instance of each of the following type occupy (at least)?

(1) BBS

(2) POOBBS

(3) NetBBS

(4) PTTCasino

(5) BBSBoardSystem

```
1  public class BBSDemo{
2    public static void main(String[] argv){
3      PTTBBS ptt1 = new PTTBBS();
4      PTTBBS ptt2 = new PTTBBS();
5      NetBBS net = new NetBBS();
6      BBS bbs = ptt2;
7      net.ptt = ptt1;
8      net.poo = new PPPBBS();
9      ptt1.casino = new POOCasino();
10
11     /* CODE */
12   }
13 }
```

Now, consider another java source file above. Please write down the output when the `/* CODE */` part is replaced by the following lines (respectively for each subproblem). If you think there is a compile error, write "compile error" or "hahaha." If you think there is a run-time error (exception), write down "run-time error" or "ohohoh."

(6) `System.out.println(ptt2 instanceof PTTBBS);`

(7) `System.out.println(ptt1 instanceof NetBBS);`

(8) `System.out.println(bbs instanceof POOBBS);`

(9) `System.out.println(net.ptt.casino instanceof PTTCasino);`

(10) `System.out.println(net.poo.casino instanceof POOCasino);`