UNIVERSITY NO. ________________________________

THE UNIVERSITY OF HONG KONG

FACULTY OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE

COMP2396/CSIS0396 Object-oriented Programming and Java

Date: 28th May, 2015 **Time: 2:30pm – 5:30pm**

Only approved calculators as announced by the Examinations Secretary can be used in this examination. It is candidates' responsibility to ensure that their calculator operates satisfactorily, and candidates must record the name and type of the calculator used on the front page of the examination script.

Please write down your university no. on top of every page.
Answer **ALL** questions (100%)

UNIVERSITY NO. ______________________________

Section A (40%) (Please circle the correct answer.)

1. How many objects are created inside the test() method?

```
public class Test {
  public static void main(String[] args) {
    Test t = new Test();
    t.test();
  }
  public void test() {
    Integer[] intArray = new Integer[3];
    intArray[0] = 1;
    intArray[1] = 2;
  }
}
```

a) 1　　b) 2　　c) 3

d) 4　　e) 5

2. What is the output of the following program?

```
public class TestInt {
  private int x;
  public TestInt(int x) { this.x = x; }
  public static void main(String[] args) {
    TestInt[] test = new TestInt[6];
    test[0] = new TestInt(1);
    test[1] = new TestInt(1);
    test[2] = new TestInt(2);
    test[3] = test[2];
    int count = 0;
    for (int i  = 1; i < test.length; ++i) {
      if (test[i - 1] == test[i]) { count++; }
    }
    System.out.println(count);
  }
}
```

a) 1　　b) 2　　c) 3

d) 4　　e) The program fails at runtime

3. To make a method accessible only within the class it belongs and classes in the same package, which of the following access level(s) should be used?

i. public

ii. default

iii. protected

iv. private

a) i or iii　　b) ii or iii　　c) ii

d) iii　　e) iv

4. What is the output of the following program?

```
public class TestA {
  static int x;
  static TestA a;
  static TestB b;
  TestA() { x++; }
  static void runTest() {
    a = new TestA();
    b = new TestB();
  }
  public static void main(String[] args) {
    runTest();
    runTest();
    System.out.println(a.x + b.x);
  }
}

class TestB extends TestA {
  static int x;
  TestB() { x+= 2; }
}
```

a) 2 b) 4 c) 6

d) 8 e) None of the above

5. What is the output of the following program?

```
public class TestThrow extends Exception {
  public static void main(String[] args) throws Exception {
    System.out.println(new TestThrow().test());
  }
  public String test() throws Exception {
    try { throw new TestThrow(); }
    catch (Exception e) { return test2(); }
    finally { return "B"; }
  }
  public String test2() throws Exception {
    try { throw new TestThrow(); }
    catch (Exception e) { throw new TestThrow(); }
    finally { return "A"; }
  }
}
```

a) A b) B c) Exception is thrown

d) This is no output e) The program fails to compile

6. Consider the following Java program:

```
public class Java extends RuntimeException {
  static Java java;
  public Java() { }
  public Java(Java java) { this.java = java; }
  public Java java() { return java; }
  public static void main(String[] args) {
    // ...
  }
}
```

Which of the following statement is/are invalid in the `main()` method?

```
i.   Java java = new Java();
ii.  Java javajava = new Java().java.java;
iii. throw new Java();
iv.  Java.java = new Java();
```

a) ii only   b) ii & iii   c) The program fails to compile

d) ii, iii & iv   e) None of the above

7. What is the output of the program?

```
public class Base {
  int i = 100;
  Base() { System.out.println(i); }
}

public class Pri extends Base {
  static int i = 200;
  public static void main(String[] args) {
    Pri p = new Pri();
    System.out.println(i);
  }
}
```

```
a) 100          b) 100          c) 200
   200                             100
d) 200          e) None of the above
```

8. What is the output of the following code segment?

```
for (int i = 1; i >= -1; --i) {
  try {
    int a = 1/i;
    System.out.println(a);
  } catch (Exception e) { System.out.println("error"); }
}
```

a) 1   b) 1error   c) The program fails to compile

d) 1error-1   e) None of the above

9. Consider the following Java program:

```
class X { }

class Y extends X { }

class Z extends Y { }

public class TestXYZ {
  public static void main(String[] args) {
    X x = new X();
    Y y = new Y();
    Z z = new Z();
    // (1) insert a statement here ...
  }
}
```

Which of the following statements, if inserted into (1), will make the program fail to compile?

```
i.   x = y;
ii.  z = x;
iii. y = (Z) x;
iv.  z = (Z) y;
```

a) i b) ii c) ii & iii

d) iii & iv e) ii, iii & iv

10. Which of the following statements is/are **NOT** correct?

i. All variables live on the stack.

ii. All objects live on the heap.

iii. A local variable is alive only when its stack frame is at the top of the stack.

iv. An object becomes eligible for garbage collection when a reference to this object is set to null.

a) i b) ii & iv c) i & iii

d) i, iii, iv e) iv

11. Which of the following statements is/are correct?

i. A sub-class can inherit a private instance variable from its super-class.

ii. A sub-class can inherit a constructor from its super-class.

iii. A sub-class can override all inheritable methods from its super-class.

iv. A sub-class can override an instance variable inherited from its super-class.

a) i b) ii & iii c) iii

d) iii & iv e) None of the above

Refer to the following Java code for questions 12 – 14.

```
public abstract class C {
  public abstract void m1(int i);
  public abstract void m1(double d);
  public void m2() { System.out.println("C:m2"); }
}

public class C1 extends C {
  public void m1(int i) { System.out.println("C1:m1i"); }
  public void m1(double d) { System.out.println("C1:m1d"); }
  public void m2() { System.out.println("C1:m2"); }
}

public class C2 extends C1 {
  public void m1(int i) { System.out.println("C2:m1i"); }
  public void m2() { System.out.println("C2:m2"); }
}
```

12. Which of the following statements will cause an error?

i. `ArrayList<C> a = new ArrayList<C>();`

ii. `a.add(new C());`

iii. `a.add(new C1());`

iv. `a.add((C) (new C2()));`

a) i b) ii c) i & ii

d) ii & iii e) iii & iv

13. What is the output of the following statement?

`(new C2()).m1(1f);`

a) `C2:m1d` b) `C2:m1i` c) `C1:m1d`

d) `C1:m1i` e) `C:m2`

14. Which of the following line will never be printed if class C has no other sub-classes?

a) `C1:m1i` b) `C:m2` c) `C1:m2`

d) `C2:m1i` e) `C2:m2`

Refer to the following Java code for questions 15 – 16.

```
class Animal {
  public void makeNoise() { System.out.println("Roar!"); }
}

class Dog extends Animal {
  public void makeNoise() { System.out.println("Woof!"); }
}

class Poodle extends Dog {
  public void makeNoise(int n) { System.out.println("Ruff!"); }
}
```

15. Which of the following statements will print "Woof!" to the screen?

i. `((Dog) (new Animal())).makeNoise();`

ii. `new Dog().makeNoise();`

iii. `new Poodle().makeNoise();`

iv. `((Animal) (new Dog())).makeNoise();`

a) i & ii   b) ii & iv   c) ii, iii & iv

d) ii   e) None of the above

16. When creating a new Poodle object, how many constructors will get invoked?

a) 1   b) 2   c) 3

d) 4   e) 5

17. Which of the following is/are method(s) in the class java.awt.Graphics for drawing a filled circle?

i. `drawCircle(int x, int y, int radius)`

ii. `drawOval(int x, int y, int width, int height)`

iii. `fillCircle(int x, int y, int radius)`

iv. `fillOval(int x, int y, int width, int height)`

a) i & iii   b) ii & iv   c) iii & iv

d) iii   e) iv

18. Which of the following statements is/are correct?

i. An abstract class cannot have non-abstract methods.

ii. An abstract class cannot have constructors.

iii. An abstract class cannot be instantiated.

iv. It is not legal to create a regular array of an abstract class.

a) i   b) ii   c) iii

d) i & iii   e) i, iii & iv

19. Which of the following statements is/are **NOT** correct?

i. A try block must be followed by a catch block.

ii. If you call a method that might throw a checked exception, you must wrap it in a try/catch block.

iii. Only checked exceptions can be caught.

iv. A finally block will run regardless of whether an exception is thrown.

a) i & iii b) ii c) iii

d) i, ii & iii e) iv

20. Which of the following classes can be compiled?

i.

```
class Foo1 {
  static int x;
  public void go() { System.out.println(x); }
}
```

ii.

```
class Foo2 {
  int x;
  public static void go() { System.out.println(x); }
}
```

iii.

```
class Foo3 {
  static int x;
  public static void go() { System.out.println(x); }
}
```

iv.

```
class Foo4 {
  int x;
  public static void go(int x) { System.out.println(x); }
}
```

a) i b) ii c) iii

d) i & iii e) i, iii & iv

UNIVERSITY NO. ______________________________

Section B (20%)

1. Briefly explain the terms "overriding method" and "overloading method". In particular, describe how can we define an overriding method and how can we define an overloading method. What are their differences? (10%)

2. Explain the terms "abstract class" and "interface". In particular, describe what are the similarities and differences between an abstract class and an interface, and when should we use an abstract class and when should we use an interface. (10%)

Section C (40%)

Owen would like to develop a program for a card game. In his preliminary design, there should be a Card class, a Player class and a CardGame class. Below shows an incomplete implementation of his design.

```
class Card {
  static final char[] SUITS = {'S', 'H', 'C', 'D'};
  static final char[] VALUES = {'3', '4', '5', '6', '7', '8', '9', '0',
                                'J', 'Q', 'K', 'A', '2'};
  private int suit; // 0 - 3
  private int value; // 0 - 12

  public Card(int s, int v) {
    suit = s;
    value = v;
  }
}
```

```
class Player {
  private ArrayList<Card> deck = new ArrayList<Card>();
  private Card prevCard;
  public void addCard(Card c) { deck.add(c); }
  public void play(ArrayList<Card> cardsOnTable) { }
  public int cardsInHand() { return deck.size(); }
}
```

```
class CardGame {
  private ArrayList<Card> deck;
  private ArrayList<Player> players;

  public void startGame() {
    createDeck();
    shuffleDeck()
    createPlayer(4);
    distributeCards(12);

    boolean endOfGame = false;
    ArrayList<Card> cardsOnTable = new ArrayList<Card>();
    while (!endOfGame) {
      for (Player p : players) {
        p.play(cardsOnTable);
        endOfGame = checkEndOfGame();
        if (endOfGame) break;
      }
    }
  }

  public void createDeck() { }
  public void shuffleDeck() { }
  public void createPlayers(int n) { }
  public void distributeCard(int n) { }
  public boolean checkEndOfGame() { }
}
```

1. Complete the `createDeck()` method in the `CardGame` class. In this method, you should create 52 cards and add them to the deck of cards referred to by the instance variable `deck`. (5%)

```
public void createDeck() {











}
```

2. Complete the `shuffleDeck()` method in the `CardGame` class. In this method, you should iterate through the cards in the deck. For each card, you should randomly pick another card in the deck and swap it with the current card. (5%)

```
public void shuffleDeck() {











}
```

3. Complete the `createPlayers()` method in the `CardGame` class. The argument to this method is the number of players to be created and added to the list of players referred to by the instance variable `players`. (2%)

```
public void createPlayers(int n) {







}
```

4. Complete the `distributeCards()` method in the `CardGame` class. The argument to this method is the number of cards to be distributed to each player. You should iterate through the players, remove the first card from the deck, and add it to the deck of the current player. (5%)

```
public void distributeCards(int n) {










}
```

5. Complete the checkEndOfGame() method in the CardGame class. It should return true when any of the players has no more cards in his hand. (1%)

```
public boolean checkEndOfGame() {







}
```

6. Complete the `Card` class by (i) defining getters for retrieving the suit and value of the card; (ii) overriding the `toString()` method inherited from the `Object` class for returning a textual representation of the card (i.e., "`SA`" for Ace of spades, "`H2`" for two of hearts, "`C0`" for ten of clubes and "`DK`" king of diamonds, etc.); (iii) overriding the `equals()` method (and any other necessary methods) inherited from the `Object` class such that it returns true for two cards having the same suit and same value. (5%)

7. Complete the `play()` method in the `Player` class. When the game starts with no cards on the table, the player with "three of diamonds" plays first by putting this card on the table. The next player can either put a card with a 'larger' value (3 < 4 < ... < 9 < 0 < J < Q < K < A < 2) or a card with the same value but a suit of 'higher rank' (diamonds < clubs < hearts < spades) on the table. If he does not have any of such cards, he will have to skip his turn and pass the turn to the next player. If all other players skip their turns, the player who put the last card on the table can continue by putting any of his cards on the table. Each player takes turn to play and the game continues until one of the players has no more cards in his hand. To simplify the implementation, you can simply select the first valid card in the deck to play (yes, we know this is not the best winning strategy :P). (7%)

UNIVERSITY NO. ______________________________

8. Owen would now like to develop another card game with a different set of rules and winning criteria. Please suggest how Owen can make use of the classes developed above. In particular, state which of the classes need (or do not need) to be extended, and for those which need to be extended, which of the methods should probably be overridden. Please briefly explain your answer. (5%)

9. Which of the following methods in the `CardGame` class should not be marked as private? Briefly explain your answer. (5%)

```
public void createDeck() { }
public void shuffleDeck() { }
public void createPlayers(int n) { }
public void distributeCard(int n) { }
public boolean checkEndOfGame() { }
```

END OF PAPER