COMP3021 Java Programming	**Name:** ______________________
Fall 2014	**Stu ID:** ______________________
Midterm Exam	**Email Address:** ______________________
23/10/2014	
Time Limit: 80 Minutes	

Instructions:

1. This exam contains 16 pages (including this cover page) and 4 questions.
2. This is a closed book exam.
3. Please write only in the exam paper.
4. You can use either pen or pencil.
5. Please take out your student ID card, and leave it at the top left side of your table.

Grade Table (for teacher use only)

Question	Points	Score
1	36	
2	34	
3	14	
4	16	
Total:	100	

Question 1 **[36 points]**

Please choose only **ONE** answer for each of the questions below.

(a) (4 points) Which of the following can be a correct constructor definition:

A. `public void A() {}`

B. `public A() {}`

C. `public A A() {}`

(a) **B**

(b) (4 points) Consider the following program:

```java
public class A {
  public int x;
  public A() { this.x=1; }
  public static void main (String args[]) {
    A x = new A();
    System.out.println(x.x);
}
}
```

A. The program does not compile.

B. The program compiles but has run time error as x.x has no value.

C. The program compiles and prints out 0.

D. The program compiles and prints out 1.

(b) **D**

(c) (4 points) Consider the following program:

```java
public class A {
  public int x;
  public static void main (String args[]) {
    A x = new A();
    System.out.println(x.x);
}
}
```

A. The program does not compile.

B. The program compiles but has run time error as x.x has no value.

C. The program compiles and prints out 0.

D. The program compiles and prints out 1.

(c) **C**

(d) (4 points) Consider the following program:

```
public class A {
  public int x=1;
  public A(int y) { x=y; }
  public static void main (String args[]) {
    A x = new A();
    System.out.println(x.x);
}
}
```

A. The program does not compile.

B. The program compiles but has run time error as x.x has no value.

C. The program compiles and prints out 0.

D. The program compiles and prints out 1.

(d) **A**

(e) (4 points) What is the result of running the following code?

```
1. class A {
2.   public byte getNumber () {
3.   return 1;
4.   }
5. }
6.
7. class B extends A {
8.   public short getNumber() {
9.   return 2;
10.  }
11.
12.  public static void main (String args[]) {
13.     B b = new B ();
14.     System.out.println(b.getNumber());
15.  }
16. }
```

A. 1 is printed.

B. 2 is printed.

C. An error at line 8 causes compilation to fail.

D. No error at line 8 but an error at line 14 causes compilation to fail.

(e) **C**

(f) (4 points) Consider the following program:

```
class C {
    public int x;
}

class D extends C {
    D() {}

    D(int a) {
   this();
    }

    public static void main(String[] args) {
   D x = new D(1);
   System.out.println(x.x);
    }
}
```

A. The program does not compile.

B. The program compiles but has run time error as x.x has no value.

C. The program compiles and prints out 0.

D. The program compiles and prints out 1.

(f) **C**

(g) (4 points) Given the following code, which comment line can be uncommented without introducing errors?

```
abstract class MyClass {
   protected static int i;
   private int j;
   abstract void f();
   final void g() {}
   //final void h() {} // (1)
}

final class MyOtherClass extends MyClass {
   int m;
   void f() {}
   void h() {}
   //void k() { i++; } // (2)
   //void l() { j++; } // (3)
   public static void main(String[] args) {
   MyClass mc = new MyOtherClass();
   }
}
```

A. (1)

B. (2)

C. (3)

D. None of the above.

(g) **B**

(h) (4 points) At point X in line 4, which piece of code can be added to make the code compile?

```
1. public class ExceptionTest {
2.    class TestException extends Exception {}
3.    public void runTest () throws TestException {}
4.    public void test () /* Point X*/ {
5.    runTest ();
6.    }
7. }
```

A. throws Exception

B. catch (Exception e).

C. throws RuntimeException.

D. catch (TestException e).

E. No code is necessary.

(h) **A**

(i) (4 points) Which of the following possible modifications will fix the errors in this code?

```
public class Test {
  private double code;

  public double getCode() {
    return code;
  }

  protected abstract void setCode(double code);
}
```

A. Remove abstract in the setCode method declaration.

B. Change protected to public.

C. Add abstract in the class declaration.

D. Add static in the variable declaration.

E. b and c.

(i) **C**

Question 2 [34 points]

(a) (4 points) The java.util.Date class implements java.lang.Cloneable and overrides the equals method to return true if two objects have the same date and time. Show the output of the following code.

```
import java.util.*;
public class Test extends Object {
  public static void main(String[] args) {
    Date d1 = new Date();
    Date d2 = new Date(349324);
    Date d3 = d1;
    System.out.println("(1) " + (d1 == d2));
    System.out.println("(2) " + (d1 == d3));
    System.out.println("(3) " + d1.equals(d2));
    System.out.println("(4) " + d1.equals(d3));
  }
}
```

Solution:

false
true
false
true

(b) (5 points) The following program uses an anonymous inner class to create a listener object. Rewrite it by replacing the anonymous inner class using an inner class called Button1Handler.

```
import java.util.*;
public class TestGUI extends JFrame {
  public JButton button;
  protected void buildGUI() {
    button = new JButton();
    button.addActionListener(
    new java.awt.event.ActionListener()
    {
      public void actionPerformed(java.awt.event.ActionEvent e)
      {
        // do something
      }
    }
    );
  }
}
```

Solution:

```
import java.util.*;
public class TestGUI extends JFrame
{
    public JButton button;
   // inner class definitions
   class Button1Handler implements ActionListener
   {
     public void actionPerformed(ActionEvent e)
       {
        // do something
     }
   }
   protected void buildGUI()
   {
     // initialize the buttons
     button = new JButton();
     // register an inner class action listener instance
     // for each button
     button1.addActionListener(new Button1Handler());
    }
 }
```

(c) (5 points) Show the output of running the class Test in the following code:

```
interface A {
  void print();
}

class C {}

class B extends C implements A {
  public void print() { }
}

public class Test {
  public static void main(String[] args) {
    B b = new B();
    if (b instanceof A)
      System.out.println("b is an instance of A");
    if (b instanceof C)
      System.out.println("b is an instance of C");
  }
}
```

Solution:

b is an instance of A
b is an instance of C

(d) (6 points) Show the output of running the following program:

```
class Test {
  public static void main(String[] args) {
    try {
      System.out.println("Welcome to Java");
      int i = 0;
      int y = 2/i;
      System.out.println("Welcome to Java");
    }
    catch (RuntimeException ex) {
      System.out.println("Welcome to Java");
    }
    finally {
      System.out.println("End of the block");
    }
  }
}
```

Solution:
Welcome to Java
Welcome to Java
End of the block

(e) (4 points) Correct the error in the following code, where A is a user-defined class with a no-arg constructor.

```
Object o = new A();
A a = o;
```

Solution: Object o = new A();
A a = (A)o;

(f) (4 points) What is wrong in the following code?

```java
public class Test {
  public static void main(String[] args) {
    Integer[] list1 = {1, 2};
    double[] list2 = {1.0, 2.0};
    printArray(list1);
    printArray(list2);
}

public static void printArray(Object[] list) {
  for (Object o: list)
    System.out.println(o + " ");
}
}
```

Solution: incompatible types: list2 is double[] cannot be converted to Object[] in printArray(list2).

(g) (6 points) What is the output of the following program:

```
class ExceptionA extends RuntimeException { }
class ExceptionB extends ExceptionA {}
class ExceptionC extends RuntimeException {}

class Exercise3 {
    void method1(){
   ExceptionA ex = new ExceptionB();
   try{
      int i=1;
      throw ex;
   }catch(Exception e){
      System.out.println("2");
      throw e;
   }finally{
      System.out.println("1");
   }
    }
    void method2(){
   try{ method1();
       System.out.println("4");
   }catch(ExceptionB e){
      System.out.println("5");
      throw new ExceptionC();
   }
    }
    void method3(){
   try{ method2();
   }catch(Exception e){
      System.out.println("6");
   }
    }
    public static void main(String[] args) {
   new Exercise3().method3();
   System.out.println("7");
    }
}
```

Solution: 2
1
5
6
7

Question 3 [14 points]

Write a class named Hexagon that extends GeometricObject and implements the Cloneable and Comparable interfaces. Assume all six sides of the hexagon are of equal size. The Hexagon class is defined as follows:

```
public class Hexagon extends GeometricObject
   implements Cloneable, Comparable<Hexagon> {
   private double side;
```

(a) (2 points) Construct a Hexagon with the specified side.

```
public Hexagon(double side) {
```

Solution:

```
this.side = side;
```

```
}
```

(b) (2 points) Implement the abstract method getArea in GeometricObject. (Hint: $area = 3 * \sqrt{3} * side * side$ and `Math.squr(double x)` is a Java library function that returns the square root of `x`.

```
public double getArea() {
```

Solution:

```
return 3 * Math.squr(3) * side * side;
```

```
}
```

(c) (2 points) Implement the abstract method getPerimeter in GeometricObject.

```
public double getPerimeter() {
```

Solution:

```
return 6 * side;
```

```
}
```

(d) (5 points) Implement the compareTo method in the Comparable interface: `obj1.compareTo(obj2)` return 0 if the two hexagons have the same area, 1 if `obj1`'s area is larger, and -1 otherwise.

```
public int compareTo(Object obj) {
```

Solution:

```
if (this.side > obj.side)
      return 1;
else if (this.side == obj.side)
      return 0;
else
      return -1;
```

```
}
```

(e) (3 points) Implement the clone method in the Cloneable interface so that X.Clone() will return a new hexagon with the same size as X.

```
public Hexagon clone() {
```

Solution:

```
  return new Hexagon(this.side);
or
  return (Hexagon)super.clone();
```

```
}
```

Question 4 **[16 points]**

A boolean expression is constructed using boolean values *true* and *false*, and boolean operators "¬" (negation) and "∧" (conjunction). Consider the following partial implementation of these booleans expressions in Java:

```java
abstract class BE {
  public abstract boolean result();
}

class Const {
  private boolean value;
  public boolean result() { return value; }
}

//Binary Boolean Expressions

abstract class BBE {
 protected BE left;
 protected BE right;
 public abstract boolean result();
}

//Unary Boolean Expressions

abstract class UBE {
 protected BE operand;
 public abstract boolean result();
}

// Negation

class Negation {
  public boolean result() { return !operand.result(); }
}

// Conjunction

class Conjunction {
  public boolean result() { return (left.result() && right.result());}
}
```

(a) (12 points) Please complete the constructor and inheritance definition for each of the above classes where necessary.

Solution:

```
abstract class BE {
  public abstract boolean result();
}

class Const extends BE { // a subclass of BE
  private boolean value;
    Const(boolean value) { this.value = value; } //contructor
  public boolean result() { return value; }
}

//Binary boolean expressions
abstract class BBE extends BE { // a subclass of BE
 protected BE left;
 protected BE right;
 public abstract boolean result();
}

//Unary boolean expressions
abstract class UBE extends BE { // a subclass of BE
 protected BE operand;
 public abstract boolean result();
}

// negation
class Negation extends UBE { // a subclass of UBE
    Negation(BE operand) { this.operand = operand; } //constructor
  public boolean result() { return !operand.result(); }
}

class Conjunction extends BBE { // a subclass of BBE
    Conjunction(BE left, BE right) { // constructor
   this.left=left; this.right=right; }
  public boolean result() { return (left.result() && right.result());}
}
```

(b) (4 points) Please give a Java code for computing the truth value of the following expression:

$$\neg(true \wedge \neg(false \wedge true)).$$

Solution:

```
class Q4 {
    public static void main(String[] args) {
   System.out.println(new Negation(new Conjunction(new Const(true),
         new Negation(new Conjunction(new Const(false),
            new Const(true))))).result());
    }
}
```