

CPSC 233 Final Exam, Winter 2003

DEPARTMENT OF COMPUTER SCIENCE

THE UNIVERSITY OF CALGARY

Time: 120 minutes

100 marks total

L02, L03, L04

First Name _____ Last Name _____

Check the box to indicate the lecture that you are registered in:

L02: MWF 10:00 – 10:50

L03: MWF 14:00 – 14:50

L04: TR 8:00 – 9:15

This is a closed book exam.

No calculators or other computational devices are allowed.

Unless otherwise indicated assume that all programs and code fragments *will* compile.

Manage your time carefully otherwise you may not be able to finish this exam.

For the multiple choice section mark your answers on the **bubble sheet**.

For the code traces and code writing write all your answers in this **exam booklet**.

First name _____ Last name _____ ID _____

Section	Your score	Max score
Multiple choice		20
Code traces		37
Code writing		29
Parsing		14
TOTAL		100

Section I, Multiple choice: 20 marks

For each question select the option that provides the *best* answer to that question.

1. Which of the following statements are true of the version of Java currently available on the Computer Science network?
 - a. “javac” is the Java compiler
 - b. “java” is the Java interpreter and is invoked as you run your java program
 - c. “javadoc” is used to produce documentation for your java code
 - d. All of the above are true
 - e. None of the above are true

For questions 2 – 4 please refer to the following program:

```
class Parent
{
    private int num1;
    protected int num2;
    public int num3;

    // Accessory and mutator methods
    public void setNum1 (int n) { num1 = n; }
    public int getNum1 () { return num1; }
    public void setNum2 (int n) { num2 = n; }
    public int getNum2 () { return num2; }
    public void setNum3 (int n) { num3 = n; }
    public int getNum3 () { return num3; }

    public void methodOne ()
    {
    }
}
}
```

```

class Child extends Parent
{
    private int num4;
    public void setNum4 (int no) { num4 = no; }
    public int getNum4 () { return num4; }

    public void methodTwo ()
    {
    }
}

class Driver
{
    public static void main (String [] argv)
    {
        Parent p = new Parent ();
        Child c = new Child ();
    }
}

```

2. Which of the following is true about the field num1?
 - a. It can be directly accessed in methodOne e.g., num1 = 1;
 - b. It can be directly accessed in methodTwo e.g., num1 = 1;
 - c. It can be directly accessed in main e.g., p.num1 = 1;
 - d. (a) & (b)
 - e. All of the above are true

3. Which of the following is true about the field num3?
 - a. It can be directly accessed in methodOne e.g., num3 = 3;
 - b. It can be directly accessed in methodTwo e.g., num3 = 3;
 - c. It can be directly accessed in main e.g., p.num3 = 3;
 - d. (a) & (b)
 - e. All of the above are true

4. Which of the following is true about the field num4?
 - a. It can be directly accessed in methodOne e.g., num4 = 4;
 - b. It can be directly accessed in methodTwo e.g., num4 = 4;
 - c. It can be directly accessed in main e.g., c.num4 = 4;
 - d. (a) & (b)
 - e. All of the above are true

For questions 5 & 6 please refer to the following program.

```

class Data
{
    private int num = 0;
    public Data ()
    {
        num = 1;
    }
}

```

```

public Data (int n)
{
    num = n;
}
public int getNum () { return num; }
public void setNum (int n) { int num = n; }
public String toString ()
{
    String s = new String ();
    s = s + num;
    return s;
}
}

class Q5
{
    public static void main (String [] argv)
    {
        Data d1 = new Data ();
        Data d2 = new Data (10);
        System.out.println("#1: " + d1 + " " + d2);
        d1.setNum(1967);
        d2.setNum(1967);
        System.out.println("#2: " + d1 + " " + d2);
    }
}

```

5. What will be the output of the first println: System.out.println("#1: " + d1 + " " + d2)?
 - a. #1: 0 0
 - b. #1: 1 1
 - c. #1: 1 10
 - d. #1: 10 10
 - e. None of the above are true

6. What will be the output of the second println: System.out.println("#2: " + d1 + " " + d2)?
 - a. #2: 1 10
 - b. #2: 1 1967
 - c. #2: 1967 10
 - d. #2: 1967 1967
 - e. None of the above are true

7. What type of relationship exists between classes "A" and "B".

```

class A
{
}

class B extends A
{
}

```

- a. An “is-a” relation
 - b. Composition
 - c. Inheritance
 - d. (a) & (c)
 - e. A “has-a” relation
8. A null pointer exception:
- a. Is an example of an unchecked exception
 - b. Is an example of a checked exception
 - c. Must be handled within a try-catch block
 - d. Must be handled within a try-catch-finally block
 - e. None of the above

For questions 9 – 12 please refer to the following code fragment.

```
class Tam
{
    private int num1 = 1;
    private static int num2 = 2;
    private static final int num3 = 3;
    void fun ()
    {
        int num4 = 4;
    }
}
```

9. Which of the following are the class variables of class Tam?
- a. num1
 - b. num2
 - c. num3
 - d. num4
 - e. All of the above are class variables
10. Which of the following are constants of class Tam?
- a. num1
 - b. num2
 - c. num3
 - d. num4
 - e. All of the above are constants
11. Which of the following are instance fields of class Tam? This means that separate memory for this field is allocated for each instance of the class.
- a. num1
 - b. num2
 - c. num3
 - d. num4
 - e. All of the above are instance fields

12. Which of the following are local variables?
- a. num1
 - b. num2
 - c. num3
 - d. num4
 - e. All of the above are local variables
13. Which of the following statements are true about parameter passing in Java?
- a. Simple, built-in, types like integers are passed by value
 - b. Simple, built-in, types like integers are passed by reference
 - c. References to objects are passed by value
 - d. (a) & (c)
 - e. None of the above

For questions 14 – 16 please refer to the following program.

```
class Q14
{
    public static void swap1 (int num1, int num2)
    {
        num1 = num2;
        num2 = num1;
    }

    public static void swap2 (int num1, int num2)
    {
        int temp;
        temp = num1;
        num1 = num2;
        num2 = temp;
    }

    public static void main (String [] argv)
    {
        int num1 = 1;
        int num2 = 2;
        System.out.println("#1: " + num1 + " " + num2);
        swap1(num1,num2);
        System.out.println("#2: " + num1 + " " + num2);
        swap2(num1, num2);
        System.out.println("#3: " + num1 + " " + num2);
    }
}
```

14. What is the output of the first println: System.out.println("#1: " + num1 + " " + num2)?
- a. #1: 0 0
 - b. #1: 1 1
 - c. #1: 1 2
 - d. #1: 2 1
 - e. None of the above

15. What is the output of the second println: System.out.println("#2: " + num1 + " " + num2)?

- a. #2: 1 1
- b. #2: 1 2
- c. #2: 2 1
- d. #2: 2 2
- e. None of the above

16. What is the output of the third println: System.out.println("#3: " + num1 + " " + num2)?

- a. #3: 1 1
- b. #3: 1 2
- c. #3: 2 1
- d. #3: 2 2
- e. None of the above

17. What will be the output of the following program?

```
public class Q17
{
    public static void main (String [] argv)
    {
        int num1, num2, num3;
        num1 = 34;
        num2 = num1++;
        num3 = ++num1;
        System.out.println(num1 + " " + num2 + " " + num3);
    }
}
```

- a. 34 34 34
- b. 36 34 35
- c. 36 35 36
- d. 36 34 36
- e. None of the above

18. Which of the following statements are accurate descriptions of instantiation in Java?

- a. Instantiation is the act of defining a class (what data and methods does the class consist of).
- b. Instantiation is when memory is dynamically allocated for an instance of a class
- c. Instantiation is physically impossible in Java
- d. (a) & (b)
- e. Instantiation is possible in web-based Java programs but not in the programs that you wrote for CPSC 233 this semester.

19. Which of the following statements is true about Java programs?
- a. Overriding refers to methods with the same name & parameter list but have separate definitions in the parent and child class.
 - b. Overloading refers to methods that have the same name but are distinguished by their parameter lists.
 - c. Preceding a data field with the word private in a class definition is an example of information/data hiding e.g., `class Foo { private int num; }`
 - d. One of the main differences between classes and interfaces is that classes have behavior (method definitions) whereas interfaces do not.
 - e. All of the above are true
20. What world famous University of Calgary graduate is widely credited with overseeing the design and implementation of the Java programming language?
- a. James Parker
 - b. James Tam
 - c. James Gosling
 - d. James Bond
 - e. Don Bidulock

Section II, Code traces: 37 marks

Questions 1 & 2 involve the tracing of Java code while you are to trace the output of a C program for Question 3.

Question 1, **Java code** (14 marks):

```
class Person
{
    protected float height;
    protected float weight;
    protected String name;

    public Person ()
    {
        height = 0;
        weight = 0;
        name = "The person with no name";
    }

    public Person (float h, float w, String n)
    {
        height = h;
        weight = w;
        name = n;
    }

    public float getHeight () { return height; }
    public void setHeight (float h) { height = h; }
    public float getWeight () { return weight; }
    public void setWeight (float w) { weight = w; }
    public String getName () { return name; }
    public void setName (String n) { name = n; }
}

class Athlete extends Person
{
    protected int heartRate;
    public Athlete ()
    {
        super();
        height = 69;
        weight = 150;
        heartRate = 50;
        name = "Fit person";
    }

    public void workOut ()
    {
        heartRate = heartRate + 1;
        System.out.println(name + " at " + heartRate + " beats per
            minute");
    }

    public void workOut (int level)
    {
        heartRate = heartRate * level;
        System.out.println(name + " at " + heartRate + " beats per
            minute");
    }
}
```

```
class Surfer extends Athlete
{
    public Surfer ()
    {
        heartRate = 40;
        name = "Board Master";
    }

    public Surfer (int h, int w)
    {
        height = h;
        weight = w;
        name = "Big Board Master";
        heartRate = heartRate * 2;
    }

    public void goSurf ()
    {
        heartRate = heartRate + 1;
    }

    public void workOut ()
    {
        goSurf();
        System.out.println(name + " at " + heartRate + " beats per
            minute");
    }

    public void workOut (int level)
    {
        heartRate = heartRate * (level+1);
        System.out.println(name + " at " + heartRate + " beats per
            minute");
    }
}

class Golfer extends Athlete
{
    public Golfer ()
    {
        super();
        heartRate = 72;
        name = "Golf lover";
    }

    public Golfer (String n)
    {
        super ();
        heartRate = 68;
        name = n;
    }

    public void workOut (int level)
    {
        heartRate = heartRate * (level-1);
        System.out.println(name + " at " + heartRate + " beats per
            minute");
    }
}
```

```
public void workOut (int level, String n)
{
    int heartRate = level;
    name = n;
    heartRate = heartRate * level;
    System.out.println(name + " at " + heartRate + " beats per
        minute");
    System.out.println("The over-indulgence in physical activity is
        not recommended by this university");
}
}

class Driver
{
    public static void main (String [] argv)
    {
        Surfer jt = new Surfer (69,155);
        Golfer dsb = new Golfer ("Tiger x 2");
        jt.workOut(1);
        dsb.workOut(2);
        jt.workOut();
        dsb.workOut(3);
        jt.workOut(3);
        dsb.workOut(10, "Super Jock");
    }
}
```

<< Put your answer here >>

<< End answer space >>

Question 2, **Java code** (15 marks):

You can assume that the code for terminal input/output library (tio) will run without problems in this example Java program. What will be the output for the program *immediately* after the person types in the following series of numbers:

```
Enter non-negative integer: 1
Enter non-negative integer: 2
Enter non-negative integer: 3
Enter non-negative integer: 4
Enter non-negative integer: 5
Enter non-negative integer: 6
Enter non-negative integer: 7
Enter non-negative integer: 8
Enter non-negative integer: 9
Enter non-negative integer: -1
```

```
import tio.*;

class Driver
{
    public static void main (String [] argv)
    {
        int number;
        IntegerList list = new IntegerList ();
        System.out.println("Enter a list of 14 non-negative integers");
        System.out.println("or a negative value to enter less than
            fourteen.");
        try
        {
            for (int i = 0; i < 14; i++)
            {
                System.out.print("Enter non-negative integer: ");
                number = Console.in.readInt();
                list.addToEnd(number);
            }
        }
        catch (EndOfListException e)
        {
            System.out.println("End of list reached.");
        }
        finally
        {
            list.display();
        }
    }
}

class IntegerList
{
    public static final int SIZE = 14;
    private int [] list;
    private int index;
    public IntegerList ()
    {
        int i;
        index = 0;
        list = new int [SIZE];
        for (i = 0; i < SIZE; i++) list[i] = 888;
    }
}
```

```
public void addToEnd (int no) throws EndOfListException
{
    if (no < 0)
        throw new EndOfListException ();

    list[index] = no;
    index++;
}

public void display ()
{
    int i;
    for (i = 0; i < SIZE; i++)
        System.out.println(list[i]);
}
}

class EndOfListException extends Exception
{
    public EndOfListException ()
    {
    }

    public EndOfListException (String s)
    {
        super(s);
    }
}
```

<< Put your answer here >>

<< End answer space >>

Question 3, C code (8 marks):

```
void
fun1 (char *ptr1, char * ptr2)
{
    char temp = *ptr1;
    *ptr1 = *ptr2;
    *ptr2 = temp;
}

void
fun2 (char c1, char c2)
{
    char k = 48;
    c1 = c1 | k;
    c2 = c2 & k;
    printf("%d %d %c %c\n", c1, c2, c1, c2);
}

int
main ()
{
    char c1, c2;
    c1 = '6';
    c2 = '*';
    printf("%d %d\n", c1, c2);
    fun1(&c1, &c2);
    printf("%d %d\n", c1, c2);
    fun2(c1, c2);
    return(0);
}
```

<< Put your answer here >>

<< End answer space >>

Section III, Code writing: 29 marks

For the first question you are to write Java code. For the second question you are to write your code in C.

Question 1, **Java code** (17 marks):

For the following question please refer to the Java code for class Person and class Instructor. Write the code for class Prof, which is a child of class Instructor. Your code for class Prof must include the following abilities:

- Create a no-argument constructor that will initialize the salary field of an instance of a Prof to \$30,000.
- Overload the constructor so that it takes one parameter, an integer “sal”. The value of the salary field will be set to sal.
- Override the “toString” method so that whenever an instance of a Prof is passed to println it will display the message “I am a Prof!”.

```
class Person
{
    private float height;
    private float weight;
    private String name;

    public Person ()
    {
        height = 0;
        weight = 0;
        name = "The person with no name";
    }

    public Person (float h, float w, String n)
    {
        height = h;
        weight = w;
        name = n;
    }

    public String toString ()
    {
        return("I am a person!");
    }

    public float getHeight () { return height; }
    public void setHeight (float h) { height = h; }
    public float getWeight () { return weight; }
    public void setWeight (float w) {weight = w; }
    public String getName () { return name; }
    public void setName (String n) {name = n;}
}

class Instructor extends Person
{
    protected float salary;
    public Instructor ()
    {
        super();
        salary = 20000;
    }
}
```

```
public String toString ()
{
    return("I am an Instructor!");
}
public float getSalary () { return salary; }
public void setSalary (float s) { salary = s; }
}
```

<< Put your answer here >>

<< End answer space >>

Question 2, **C code** (12 marks):

Write a C program that will read in any number of octal numbers (integers) from the command line and displays one of the following outputs:

- Their equivalent decimal value (as an integer value) – value can be converted.
- An error message “Wrong format for octal number” – value in the wrong format and cannot be converted.

In both cases the output will be followed by a newline. For example, if the user of the program typed in the following inputs at the command line:

```
./a.out 1 2 3 4 5 6 7 8 10 11 12
```

The program would generate the following output:

```
1
2
3
4
5
6
7
Wrong format for octal number
8
9
10
```

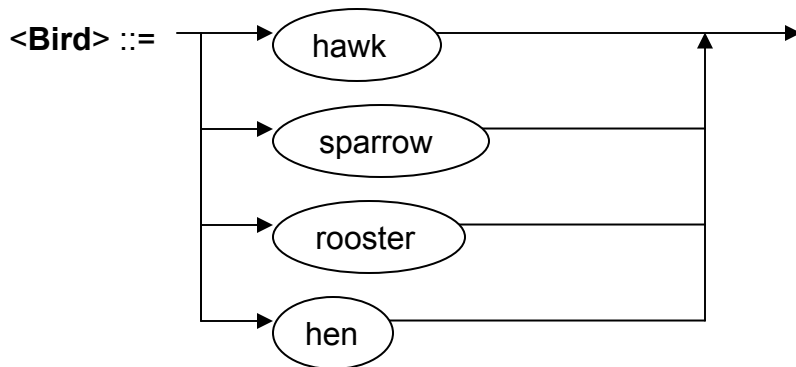
<< Put your answer here >>

<< End answer space >>

Section IV: Parsing, BNF and Syntax Diagram: 14 marks

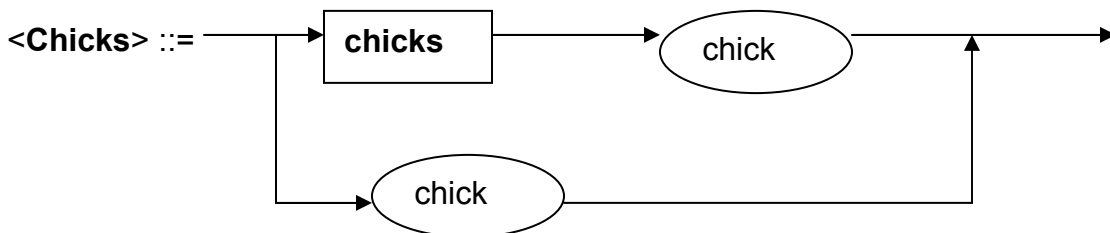
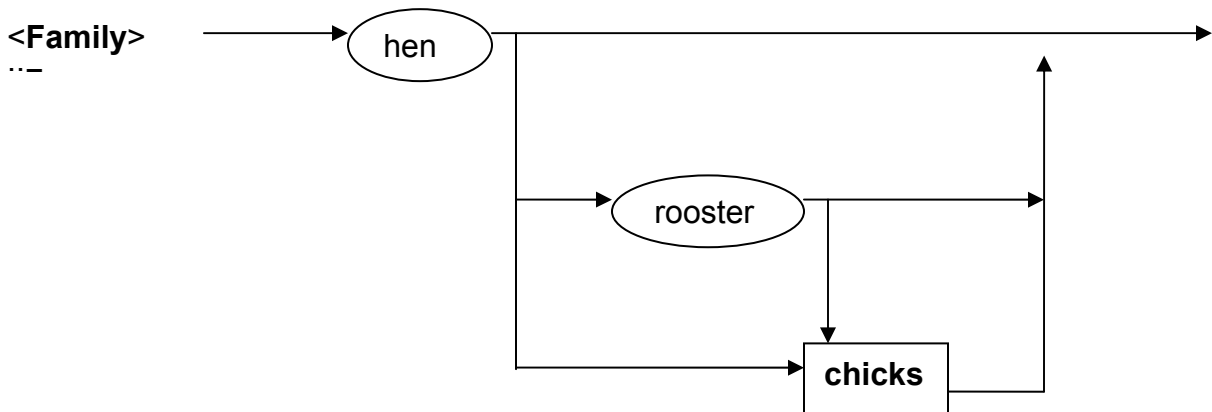
Please refer to the syntax diagrams and BNF specifications in order to answer the questions for this section.

<Bird> ::= <hawk> | <sparrow> | <rooster> | <hen>



<Family> ::= <hen> | <hen> <rooster> | <hen> <chicks> | <hen> <rooster> <chicks>

<Chicks> ::= <chicks> <chick> | <chick>



Question 1, **Parsing** (2 marks):

Is a hawk a bird? Is a hen a bird?

Question 2, **Parsing** (2 marks):

Is a quail a bird? Is a chick a bird?

.

For questions 3 – 8 you will be given some candidate families or candidate chicks. You must indicate whether or not these candidate families/chicks follow the rules defining a family/chicks according to the BNF notations and the syntax diagrams. If the answer is no, then you need to back it up with **all** the instances where the syntax has been violated.

Question 3, **Parsing** (1 mark):

Candidate family: hen

Question 4, **Parsing** (1 mark):

Candidate family: hen rooster chick chick

Question 5, **Parsing** (2 marks)

Candidate family: rooster chick

Question 6, **Parsing** (2 marks)

Candidate family: hen chick rooster

Question 7, **Parsing** (3 marks)

Candidate family: chick rooster rooster hen

Question 8, **Parsing** (1 mark)

Candidate chicks: <nothing>