CS 2334: Programming Structures and Abstractions: Exam 1
October 3, 2016

General instructions:

- Please wait to open this exam booklet until you are told to do so.
- This examination booklet has 13 pages. You also have been issued a bubble sheet.
- Fill in the identifying information below (signature, name, ID and date) Also, write your name and ID number on your bubble sheet, and fill in the bubbles for your ID.
- You may have up to five pages of your own notes. No electronic devices or books may be used.
- The exam is worth a total of 137 points. Your grade counts for 10% of your final grade.
- You have 1.25 hours to complete the exam. Be a smart test taker: if you get stuck on one problem go on to the next.
- Use your bubble sheet to answer all multiple-choice questions. Make sure that the question number and the bubble row number match.
- Other than **this** page, you may tear any other page out of this booklet that does not contain numbered answers.
- If you cannot effectively erase erroneous answers from the bubble sheet, please clearly cross them out.

On my honor, I affirm that I have neither given nor received inappropriate aid in the completion of this exam.

Signature: _____ **Name:** _____

ID Number: _____ **Date:** _____

C A B C C A C B A C B

Question	Points	Score
Types and Objects	32	
Inheritance and Polymorphism	35	
UML and Object Oriented Design	14	
Abstract Classes and Interfaces	24	
Exceptions and Error Handling	28	
Miscellaneous	4	
Total:	137	

Part I. Types and Objects

1. (4 points) What is printed by this block of code?

```
1  int i1 = 7;
2  int i2 = 5;
3  System.out.println(i1/i2);
```

A. 0 B. 1 C. 1.4 D. 12 E. Compilation error or answer not shown

2. (4 points) What is printed by this block of code?

```
1  int a = 4;
2  double c = 11;
3  int b = c+7;
4  c += b;
5  System.out.println(c);
```

A. 15.0 B. 18.0 C. 22.0 D. 29.0
E. Compilation error or answer not shown

3. (4 points) What is printed by this block of code?

```
1  String a = "OOM";
2  Integer b = -5;
3  Integer c = new Integer(19);
4  System.out.println(c + b + a);
```

A. 14OOM B. 19-5M C. 19-5OOM D. -519OOM
E. Compilation error or answer not shown

4. (4 points) What is printed by this block of code?

```
1  int i1 = 42;
2  String s = "8";
3  System.out.println(i1 + s);
```

A. 6 B. 50 C. 428 D. 842 E. Compilation error or answer not shown

5. (4 points) What is printed by this block of code?

```
String s1 = "XHT";
String s2 = "xHt";
s2.toUpperCase();
if(s1.equals(s2))
{
  System.out.println("Yes:" + s1);
}
else
{
  System.out.println("No:" + s2);
}
```

A. No:xHt B. No:XHT C. Yes:xHt D. Yes:XHT
E. Compilation error or answer not shown

6. (4 points) What is printed by this block of code?

```
int a = 17;
String b = "YYO";
int c = 22;
System.out.println(b + c + a);
```

A. YY39 B. YYO39 C. YYO1722 D. YYO2217
E. Compilation error or answer not shown

7. (4 points) What is printed by this block of code?

```
int a = 3;
int b = 7;
String c = "7117";
System.out.println(c + (a + b));
```

A. 7127 B. 711710 C. 711737 D. 711773
E. Compilation error or answer not shown

8. (4 points) What is printed by this block of code?

```
int a = 32;
String b = "4EB";
int c = 8;
System.out.println(a + c + b);
```

A. 324EB8 B. 3284EB C. 404EB D. 84EB32
E. Compilation error or answer not shown

Part II. Inheritance and Polymorphism

Consider the following class definitions:

```
public class A
{
  private int val;

  public A(int val) {
    this.val = val;
  }

  public int getVal() {
    return val;
  }

  public String toString() {
    return "A:" + this.getVal();
  }
}

public class B extends A
{
  private String name;

  public B(int val, String name) {
    super(val);
    this.name = name;
  }

  public String getName() {
    return name;
  }
}

public class C extends B
{
  private int val;

  public C(int val, String name) {
    super(val*2, name);
    this.val = val;
  }

  public C(String name, int val) {
    super(val, name.toLowerCase());
    this.val = -1;
  }

  public int getVal() {
    return val;
  }

  public int getSuperVal() {
    return super.getVal();
  }

  public String toString() {
    return "C:" + this.getName() + ":" + super.toString();
  }
}
```

9. (7 points) What is printed by this block of code?

```
A a = new A(42);
System.out.println(a);
```

A. A: B. A:42 C. A:84 D. A:A:42 E. Answer not shown

10. (7 points) What is printed by this block of code?

```
B b = new B(79, "Bob");
System.out.println(b);
```

A. A:79 B. B:79 C. Bob:79 D. B:Bob:79 E. Answer not shown

11. (7 points) What is printed by this block of code?

```
C c = new C(83, "Joe");
System.out.println(c);
```

A. C:Joe:A:-1 B. C:Joe:A:-2 C. C:Joe:A:83 D. C:Joe:166
E. Answer not shown

12. (7 points) What is printed by this block of code?

```
C e = new C("Ann", 3);
System.out.println(e.getVal() + e.getSuperVal());
```

A. -1 B. 1 C. 2 D. 3
E. Answer not shown

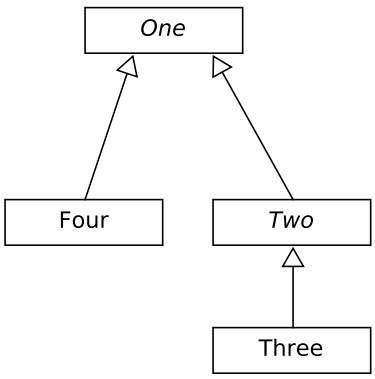
13. (7 points) What is printed by this block of code?

```
C d = new C("Henry", 19);
System.out.println(d);
```

A. C:henry:A:19 B. C:Henry:A:19 C. C:henry:A:-1 D. C:Henry:A:-1
E. Answer not shown

Part III. UML and Object Oriented Design

14. (4 points) Which set of class definitions corresponds to the following UML diagram?



A.
```
public abstract class One {...}
public abstract class Two extends One {...}
public class Three extends Two {...}
public class Four extends Two {...}
```

B.
```
public interface One {...}
public interface Two implements One {...}
public class Three extends Two {...}
public class Four implements One {...}
```

C.
```
public class One {...}
public class Two extends One {...}
public class Three extends Two {...}
public class Four extends One {...}
```

D.
```
public abstract class One {...}
public abstract class Two extends One {...}
public class Three extends Two {...}
public class Four extends One {...}
```

E. Answer not shown

15. (4 points) **Carefully examine** the following UML models and select the one that corresponds to the following code.

```
public class MyClass
{
    private double val;
    protected char c;

    public MyClass(char c, double val) {
        this.val = val;
        this.c = c;
    }

    private void print() {
        System.out.println(c);
    }

    public double doubleValue() {
        return val * 2;
    }
}
```

A.

MyClass
-val: double #c: char
+MyClass(c:char, val:double) -print(): void +doubleValue(): double

B.

MyClass
-val: double -c: char
+MyClass(c:char, val:double) -print(): void +doubleValue(): double

C.

MyClass
-c: char #val: double
+MyClass(c:char, val:double) -print(): void +doubleValue(): double

D.

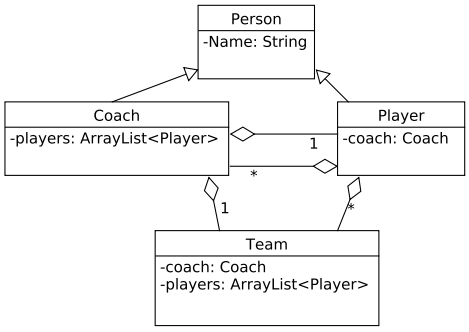
MyClass
-val: double #c: char
+MyClass(c:char, val:double) -print(): void +doubleValue(): double

E. Answer not shown

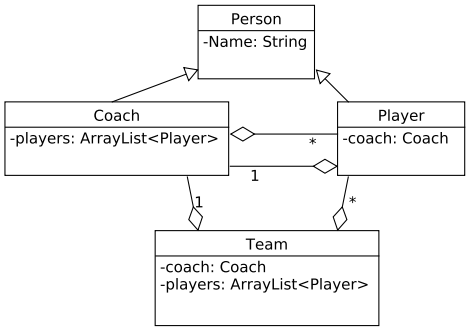
16. (6 points) Which UML diagram corresponds to the following code?

```
public class Person
{
  private String Name;
}

public class Coach extends Person
{
  private ArrayList<Player> players;
}

public class Player extends Person
{
  private Coach coach;
}

public class Team
{
  private Coach coach;
  private ArrayList<Player> players;
}
```

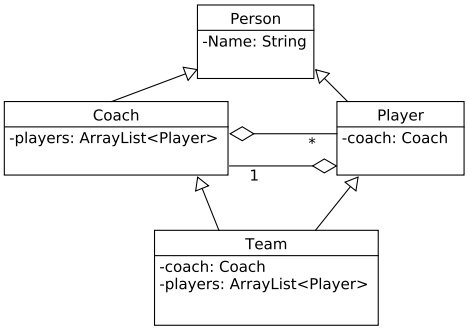
A.



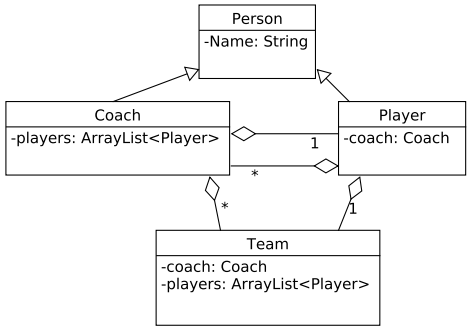
B.



C.



D.



E. Answer not shown

Part IV. Abstract Classes and Interfaces

17. (4 points) Which line (if any) will cause the program not to compile?

```
1  public interface MyList
2  {
3      public abstract void add(Integer i);
4      public abstract Integer get(int j);
5      public abstract void insert(Integer i, int j);
6  }
```

A. 1 B. 3 C. 4 D. 5 E. Answer not shown

18. (4 points) Which line (if any) will cause the program not to compile?

```
1    public interface MyInterface
2    {
3      public int value;
4
5      public abstract double setDouble(double v)
6      {
7        value = v;
8      }
9      public abstract String toString();
10   }
```

A. 3 B. 5 C. 7 D. Multiple lines
E. Answer not shown

19. (4 points) Which line (if any) will cause the program not to compile?

```
1    public abstract class Worker
2    {
3      private double workRate;
4
5      public Worker(double workRate)
6      {
7        this.workRate = workRate;
8      }
9
10     public abstract boolean canDoWork();
11
12     public double doWork()
13     {
14       if(this.canDoWork())
15       {
16         return this.workRate;
17       }
18       else
19       {
20         return 0;
21       }
22     }
23   }
```

A. 7 B. 10 C. 14 D. 16 E. Answer not shown

20. (4 points) Any class that extends an abstract class must provide implementations for all of the abstract methods.

A. True B. False C. Answer not shown

Consider the following class definition for the next two questions:

```
1   public class Course implements Comparator<Course>
2   {
3     private String name;
4     private int grade;
5
6     public Course(int name, String grade)
7     {
8       this.name = name;
9       this.grade = grade;
10    }
11
12   /**
13    * Sort first by grade and then by the
14    * natural order of the names.
15    * @return 1 if c1 comes after c2; -1 if c1 comes before c2;
16    *             and 0 if they are equal
17    */
18    public int compare(Course c1, Course c2)
19    {
20      if(c1.grade < c2.grade)
21      {
22        return -1;
23      }
24      else if(c1.grade > c2.grade)
25      {
26        return 1;
27      }
28      else
29      {
30        return c2.name.compareTo(c1.name);
31      }
32    }
33  }
```

21. (4 points) Which one line can be changed to fix a bug in the class definition or the Course constructor?

A. 1 B. 6 C. 8 D. 9 E. Answer not shown

22. (4 points) Which one line can be changed to fix a bug in the compare() method?

A. 18 B. 20 C. 24 D. 30 E. Answer not shown

Part V. Exceptions and Error Handling

Consider the following program:

```
public class ExceptionsOnExam
{
  public static int doSubJob(int value)
  {
    if(value <= 5)
    {
      throw new IllegalArgumentException("Error 1");
    }
    else if(value > 8)
    {
      throw new IllegalStateException("Error 2");
    }
    else
    {
      return 42;
    }
  }

  public static int doJob(int arg)
  {
    int value = -1;

    if (arg <= 10 && arg >= 2)
    {
      try
      {
        return doSubJob(arg + value);
      }
      catch(IllegalArgumentException e)
      {
        return value;
      }
    }
    else
    {
      return doSubJob(2 * value);
    }
  }

  public static void main(String[] args)
  {
    int value = ????;

    try
    {
      int ret = doJob(value);
      System.out.println(ret);
    }
    catch (Exception e)
    {
      System.out.println(e.getMessage());
    }
  }
}
```

Note that both *IllegalArgumentException* and *IllegalStateException* are *RuntimeExceptions*.

23. (7 points) Assume that $value = 10$ in main(), what is printed by the program?
A. Error 1 B. Error 2 C. -1 D. 42 E. Answer not shown

24. (7 points) Assume that $value = 1$ in main(), what is printed by the program?
A. Error 1 B. Error 2 C. -1 D. 42 E. Answer not shown

25. (7 points) Assume that $value = 6$ in main(), what is printed by the program?
A. Error 1 B. Error 2 C. -1 D. 42 E. Answer not shown

26. (7 points) Assume that $value = 7$ in main(), what is printed by the program?
A. Error 1 B. Error 2 C. -1 D. 42 E. Answer not shown

Part VI. Miscellaneous

Consider the following program:

```
1   public class MyClass
2   {
3     private double val;
4     protected char c;
5
6     public MyClass(char c, double val) {
7       this.val = val;
8       this.c = c;
9     }
10
11    private void print() {
12      System.out.println(c);
13    }
14
15    public double doubleValue() {
16      return val * 2;
17    }
18
19    public static void main(String[] args)
20    {
21      MyClass m = new MyClass('a', 7.2);
22      int a = 4;
23
24      System.out.println(m + a);
25    }
26  }
```

27. (2 points) In which part of memory is the variable declared on line 4 stored?
    A. Heap B. Stack C. Answer not shown

28. (2 points) In which part of memory is the variable declared on line 22 stored?
    A. Heap B. Stack C. Answer not shown