Please read Sections 9.3, and 9.5 through 9.9 of the textbook and then answer the following, trying not to look at your notes or at the textbook. Quiz #11, on Wed. 30 Nov., will consist exclusively of questions taken from the Part 1 of this homework.

Part I — Questions

Ex. 1. What is the difference, if any, between shadowing and overriding?

Ex. 2. What is the difference, if any, between overloading and overriding?

Ex. 3. What is a final method?

Ex. 4. Complete the following: "Every class inherits from the __________________ class."

Ex. 5. When you create a class, it automatically has a `toString` method and an `equals` method. Why?

Ex. 6. Assuming the `Cube` class extends the `Rectangle` class, write a statement that declares `box` as a `Rectangle` object, creates a `Cube` object using a no-arg constructor, and stores the address in the `box` variable. Does the `box` variable have access to the methods in the `Cube` class?

Ex. 7. Look at the following code:

In `Sup.java`:

```
public class Sup{
  private int y;
  public Sup(){
    y = 0;
  }
  public int getY(){
    return y;
  }
}
```

In `Sub.java`:

```
public class Sub extends Sup{
  private char a;
  public Sub(){
    super();
    a = 'c';
  }
  public char getA(){
    return a;
  }
}
```

In `Demo.java`:

```
public class Demo{
  public static void main(String[] args){
    Sup object = new Sub();
    System.out.print(object.getY());
  }
}
```

1. The reference variable `object` reference an object of type different of its own: this is called ____________.
2. What will be printed by the `main` method?
3. What would happen if we added `System.out.print(object.getY());` in the `main` method?
4. What would happen if we comment the `super();` statement in the `Sub` constructor?
5. What would happen if we added `Sub object2 = new Sup();` in the `main` method?

Ex. 8. Write the header of an abstract `toString` method.

Ex. 9. Can an abstract class be instantiated? Can it serve as a superclass?

Ex. 10. If a subclass extends a superclass that has an abstract method, then what must you do in the subclass?

Ex. 11. What is an interface made of?

Ex. 12. Can a class extends several superclasses at the same time? Can a class implements several interfaces at once?

Ex. 13. Write an interface named `ColorFul` that specifies the following methods:

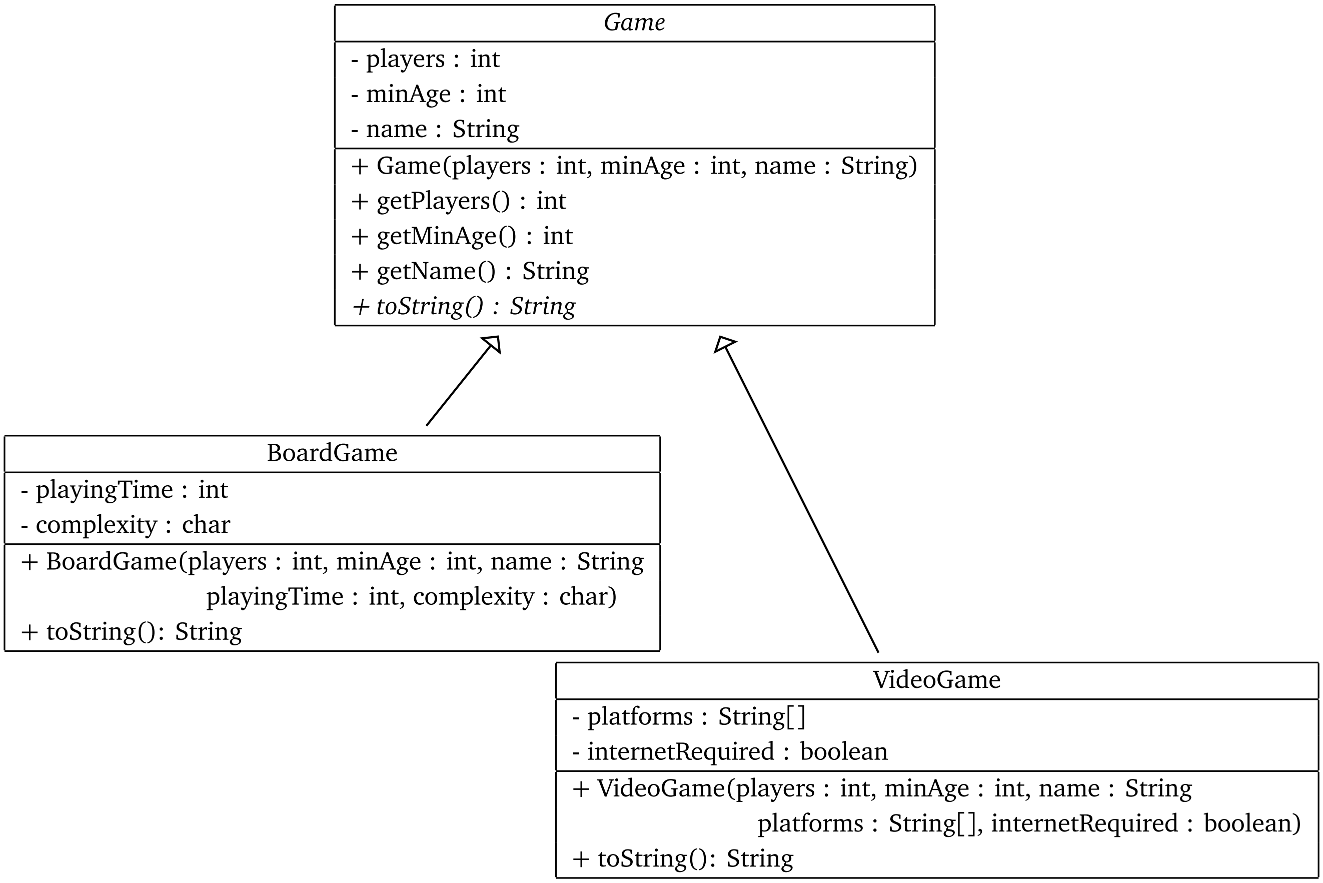
```
public void setColor(String color)
public String getName()
```

Part II — Programming Exercises

Ex. 1. We're going to develop a small program that prints lists of board and video games:

```
******** Video Game ******
The Incredible Machine
1 Players
Age 8 and +
Platforms:
        Commodore 64
        Amiga
Can be played offline.
******** Board Game ******
Carcassonne
5 Players
Age 8 and +
30+ min.
Complexity: A
******** Board Game ******
Robo Rally
8 Players
Age 12 and +
120+ min.
Complexity: B
```

Remember that classes and methods in italics in UML diagram are abstract, and implement the classes described in the following diagram:


Game
- players : int
- minAge : int
- name : String
+ Game(players : int, minAge : int, name : String)
+ getPlayers() : int
+ getMinAge() : int
+ getName() : String
+ toString() : String
BoardGame
- playingTime : int
- complexity : char
+ BoardGame(players : int, minAge : int, name : String playingTime : int, complexity : char)
+ toString(): String
VideoGame
- platforms : String[]
- internetRequired : boolean
+ VideoGame(players : int, minAge : int, name : String platforms : String[], internetRequired : boolean)
+ toString(): String

Write an application program that create `BoardGame` and `VideoGame` objects, and store them all in a single array. What should be the type of that array? Use an enhanced loop to print the content of that array. Try to print something like what was shown previous page.