Montgomery College, CMSC 203
Worksheet 1
Module 14

Objectives

- Comparing objects
- Copying objects
- Enumerated types

Concept Questions

1) You cannot use the == operator to compare the contents of:
   A) objects
   B) strings
   C) integers
   D) Boolean values

Answer: A

2) When using the == operator with two objects, only the _____ of the two objects are compared.

Answer: addresses

3) To compare two objects in a class:
   A) use the == operator, e.g. `object1 == object2`
   B) write a method to do a byte-by-byte compare of the two objects
   C) write an equals method that will make a field by field compare of the two objects
   D) Since objects consist of several fields, you cannot compare them

Answer: C

4) The two possible ways to copy objects are _____ copy and _____ copy.

Answer: Deep, Shallow

5) If `object1` and `object2` are objects of the same class, to make `object2` a deep copy of `object1`:
   A) assign `object1` to `object2`, such as `object2 = object1`;
   B) write a copy method that will make a field by field copy of `object1` data members into `object2` data members
   C) use the Java copy method that is a part of the Java language
   D) use the default constructor to create `object2` with `object1` data members

Answer: B

6) The term for the relationship created by object aggregation is:
   A) *has a*
   B) *is a*
   C) Sub-class object
   D) Inner class

Answer: A

7) A deep copy of an object:
A) is an assignment of that object to another object
B) is an operation that copies an aggregate object, and all the objects it references
C) is a bogus term, it has no meaning
D) is always a private method
Answer: B

8) A declaration for an enumerated type begins with this key word.
A) `enumerated`
B) `enum_type`
C) `enum`
D) `ENUM`
Answer: C

9) Enumerated types have this method, which returns the position of an `enum` constant in the declaration list.
A) `toString`
B) `position`
C) `ordinal`
D) `location`
Answer: C

10) Look at the following declaration:

```
enum Tree { OAK, MAPLE, PINE }
```

What is the ordinal value of the MAPLE `enum` constant?
A) 0
B) 1
C) 2
D) 3
E) Tree.MAPLE
Answer: B

11) Look at the following declaration:

```
enum Tree { OAK, MAPLE, PINE }
```

What is the fully-qualified name of the PINE `enum` constant?
A) `PINE`
B) `enum.PINE`

C) `Tree.PINE`
D) `Tree(PINE)`
E) `PINE.Tree`

Answer: C

12) Assuming the following declaration exists:

```
enum Tree { OAK, MAPLE, PINE }
```

What will the following code display?

```
System.out.println(Tree.OAK);
```

A) `Tree.OAK`
B) 0
C) 1
D) `OAK`
E) Nothing. This statement will cause an error.

Answer: D

Programming question:

Given the following book class, do the following:

```
public class Book {
	private String title;
	private String author;
	private double price;


	public Book(String title, String author, double price){
		this.title = title;
		this.author = author;
		this.price = price;
	}

}
```

1. Create a static enumerated type called `Status`. It will represent the current status of the book. Create the following statuses:
   - IN_STOCK
   - OUT_OF_STOCK
   - SHIPPED
   - DELIVERED
2. Implement an `equals` method which will be used to compare the `Book` objects. Note: the `equals` method should be an override of the method in the `Object` class.
3. Create a copy constructor which will accept an object of Book as an argument and create a deep copy of the accepted object.

4. Create the following book object:
    - Author: “Daniel”
    - Title: “Adventured of Daniel”
    - Price: 300
    - Status: OUT_OF_STOCK

5. Make a deep copy of the created book called `bookCopy`. Using the equals method check that it is indeed a copy.
6. Change the status of `bookCopy` to IN_STOCK. Using the `equals` method, make sure that both book objects are not the same.

Answer:

```
public class Book {
	static enum Status {IN_STOCK, OUT_OF_STOCK, SHIPPED, DELIVERED}

	private String title;
	private String author;
	private double price;
	private Status status;


	public Book(String title, String author, double price){
		this.title = title;
		this.author = author;
		this.price = price;

		status = Status.IN_STOCK;
	}

	public Book(Book b){
		this.title = b.title;
		this.author = b.author;
		this.price = b.price;
		this.status = b.status;
	}

	public boolean equals(Object o){
		Book b = (Book)o;
		if(b.title.equals(this.title) && b.author.equals(this.author) &&
b.price == this.price && b.status == this.status){
			return true;
		}

		return false;
	}

	public void setStatus(Status s){
		this.status = s;
	}
}
```

```java
public class Main {

	public static void main(String[] args){
		Book book = new Book("Adventures of Daniel", "Daniel", 300);
		book.setStatus(Book.Status.OUT_OF_STOCK);

		Book bookCopy = new Book(book);

		if(book.equals(bookCopy)){
			System.out.println("Books are same");
		} else {
			System.out.println("Books are not same");
		}

		bookCopy.setStatus(Book.Status.IN_STOCK);

		if(book.equals(bookCopy)){
			System.out.println("Books are same");
		} else {
			System.out.println("Books are not same");
		}

	}
}
```