Java Objects and References

CSCI 2300

In your assigned reading you learned about

- Creating objects
- Using objects
- Passing arguments to functions:
  - Strictly by value

- Any Questions?

Creating Objects Summary

- **Declare** variable: *TYPE variable_name*
- **Instantiate** class (create object) with **new** keyword
- **Initialize** object with constructor
- Primitive types: memory is allocated at declaration
- Non-primitive types (classes): memory is allocated at instantiation (when object is created with new keyword)
    - This is different from C++

CreateCircle.java

```
public class CreateCircle
{
   public static void main(String []args)
   {
      Circle2D circle;
      circle.radius = 5;
   }
}
```

What is the outcome of the main method?

Circle2D.java

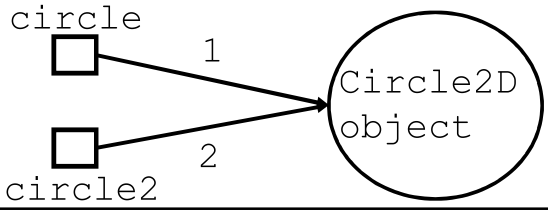
```
public class Circle2D
{
   public Point2D center;
   public int radius = 0;
}
```

A. `circle` has a radius 5
B. Error, because `center` of the `circle` has not been initialized
C. Error, because `circle` has not been instantiated
D. Error because `Circle2D` class does not have a constructor.

In Java non-primitive variables are references

- `Circle2D circle`
    - A reference to a circle object
- Multiple references to the same object

```
1. Circle2D circle = new Circle2D();
2. Circle2D circle2 = circle;
```



```
public class CirclesAndPoints
{
   public static void main(String []args)
   {
      Point2D point = new Point2D(1, 1);
      Circle2D circle1 = new Circle2D(point, 5);
      point.setX(5);
      Circle2D circle2 = new Circle2D(point, 10);
   }
}
```

What is the outcome of the main method?

```
public class Circle2D
{
   public Point2D center;
   public int radius = 0;
   public Circle2D(Point2D c, int r)
   {
      center = c;
      radius = r;
   }
}
```

Circle2D.java

A. `circle1` is centered at (1, 1) with radius 5
B. `circle1` is centered at (5, 1) with radius 5
C. `circle2` is centered at (5, 1) with radius 10
D. A and C
E. B and C

Comparing object values (incorrect)

```
import java.util.Scanner;
public class CompareStrings
{
   public static void main(String []args)
   {
      Scanner in = new Scanner(System.in);
      String name1 = in.nextLine();
      String name2 = in.nextLine();
      // compare names
      if (name1 == name2)
      {
         // ...
      }
   }
}
```

Scanner class is used for input from terminal

Always false

Why?

Comparing object values (corrected)

```
import java.util.Scanner;
public class CompareStrings
{
   public static void main(String []args)
   {
      Scanner in = new Scanner(System.in);
      String name1 = in.nextLine();
      String name2 = in.nextLine();
      // compare names
      if (name1.equals(name2))
      {
         // ...
      }
   }
}
```

Method of class String

Where can we find documentation of classes included with Java?

Using object references

- Access public methods and instance variables of an object using "dot" operator
- Example:
  `Point2D point = new Point2D(1,1);`
  `point.setX(5);`
- Garbage collector frees memory for unreferenced objects
- `point = null;`
  - `Point2D` object that was associated with `point` reference will be freed by the garbage collector

```
1. public class CreateCircle
2. {
3.    public void create(Point2D p)
4.    {
5.       Circle2D c = new Circle2D(p, 5);
6.       return;
7.    }
8.    public static void main(String []args)
9.    {
10.      Point2D p = new Point2D(1, 1);
11.      create(p);
12.      System.out.println("Created circle");
13.   }
14. }
```

Circle2D object created here is unreferenced when application reaches line 12.

Why?

Pass by Value only

- Method arguments can be:
  - Primitive type
  - Non-primitive type
- Primitive type: value of the primitive is copied as parameter
- Non-primitive type: reference is copied as parameter

Example: passing primitive types

```
public void swap(int a, int b)
{
   int temp = a;
   a = b;
   b = temp
}

public static void main(String []args)
{
   int x = 0; int y = 1;
   swap(x, y);
   System.out.println("x=" + x);
   System.out.println("y=" + y);
}
```

What is the output of this code?

Example: passing non-primitive types

```
public void swap(Point2D a, Point2D b)
{
   Point2D temp = a;
   a = b;
   b = temp
}

public static void main(String []args)
{
   Point2D x = new Point2D(1, 1);
   Point2D y = new Point2D(5, 5);
   swap(x, y);
   System.out.println("x=" + x);
   System.out.println("y=" + y);
}
```

What is the output of this code?

Example: passing non-primitive types

```
public void reset(Point2D a)
{
   a.setX(0);
   a.setY(0);
}

public static void main(String []args)
{
   Point2D x = new Point2D(1, 1);
   reset(x);
   System.out.println("x=" + x);
}
```

What is the output of this code?