

CSE8A Final Exam sample questions

The following questions are typical of the types of questions that are usually on CSE8A final exams. This is a “set” of questions. The number is not representative of the length of the exam. Nor is the number of questions on a specific topic. But the types of the questions are examples of what you will likely see.

Feel free to work these exam questions with other students and to ask TAs or the instructor to explain the answers. However, answers will not be posted on the web or emailed out. On most questions, you should be able to type in the code to figure out what the correct answer is.

X1. Describe succinctly, in English sentences, what this code does.

```
public static void main (String[] args)
{
    Scanner in = new Scanner(System.in);
    int y = 0;
    int x = in.nextInt();
    while (x>0)
    {
        if (x%3 == 0)
            y++;
        x = in.nextInt();
    }
    S.o.p(y);
}
```

X2. Write code to print out a box of stars around a box of dashes. The size of the box is determined by the user input. For example, if the user inputs 3 and 4, then the following should be output:

```
* * * *  
* _ _ *  
* * * *
```

if the user inputs 6 and 5, then the following should be output:

```
* * * * *  
* _ _ _ *  
* _ _ _ *  
* _ _ _ *  
* _ _ _ *  
* * * * *
```

X3. Write code to store a box of stars around a box of dashes into a 2-dimensional array. The size of the box is determined by the user input. For example, if the user inputs 3 and 4, then the following should be stored into an array of size just big enough to hold this box:

```
* * * *  
* _ _ *  
* * * *
```

if the user inputs 6 and 5, then the following should be stored into an array of size just big enough to hold this box:

```
* * * * *  
* _ _ _ *  
* _ _ _ *  
* _ _ _ *  
* _ _ _ *  
* * * * *
```

X4. Which of the following codes is a best default constructor for a class Foo which has instance variables as defined below.

```
public class Foo {  
  
    String cityName;  
    int temperature;  
    Date[] observes;  
  
}
```

```
A)public Foo()  
{  
    cityName = "No name";  
    temperature = 0;  
    observes = new Date();  
}
```

```
B)public Foo()  
{  
    cityName = "No name";  
    temperature = 0;  
    observes = null;  
}
```

```
C)public Foo()  
{  
    cityName = "No name";  
    temperature = 0;  
    observes = new Date[5];  
    for (int i = 0; i < observes.length; i++)  
    {  
        observes[i] = null;  
    }  
  
}
```

```
D)public Foo()  
{  
    cityName = "No name";  
    temperature = 0;  
    observes = new Date[5];  
    for (int i = 0; i < observes.length; i++)  
    {  
        observes[i] = new Date();  
    }  
}
```

X5. Write a method that implements a deep copy constructor for the `Foo` class defined on the previous page.

X6. Which of the following is a best accessor method for the `observes` instance variable that protects the privacy of the `observes` instance variable of the `Foo` class? To protect privacy, the accessor should not return an alias.

- A)

```
public Date getObserves()  
{  
    return observes[0];  
}
```
- B)

```
public int getObserves()  
{  
    return observes.length;  
}
```
- C)

```
public Date[] getObserves()  
{  
    return observes;  
}
```
- D)

```
public Date getObserves()  
{  
    return new Date(observes[0]);  
}
```
- E)

```
public Date[] getObserves()  
{  
    return new Date(observes);  
}
```
- F)

```
public Date[] getObserves()  
{  
    Date[] x = new Date[observes.length];  
    for (int i = 0; i < observes.length; i++)  
        x[i] = new Date(observes[i]);  
    return x;  
}
```

X7. Describe succinctly, in English sentences, what this code does.

```
public static double[] foo(double[] bar)
{
    if (length%2 == 0)
        double[] answer = new double[bar.length/2];
    else
        double[] answer = new double[bar.length/2+1];

    int j = 0;
    for (int i = 0; i < bar.length; i+=2)
    {
        answer[j++] = bar[i];
    }
    return answer;
}
```

X8. This code may not do what the developer intended. What does this code output when the input is:

12, 20, 28, -1

```
int count = 0;
int i = 0;
int sum = 0;
while (i > 0)
{
    sum += i;
    i = in.nextInt();
    count++;
}
```

```
System.out.println((double)sum/count);
```

- A) 20.0
- B) 18.0
- C) 15.0
- D) 12.0
- E) It has a runtime divide by zero error.

X9. What is output when this code is run?

```
Species s1 = new Species("Wombat", 200, 2.2);  
Species s2 = new Species("Wombat", 200, 2.2);
```

```
S.o.print(s1.equals(s2) + ", ");
```

```
if (s1 == s2)  
    S.o.p("Same");  
else  
    S.o.println("Not the same");
```

- A) true, Same
- B) false, Same
- C) true, Not the same
- D) false, Not the same
- E) yes, Same
- F) yes, Not the same
- G) no, Same
- H) no, Not the same
- I) A runtime error occurs at the first print statement

X10. What is output when the following code is run?

```
public class Bar
{
    public static void main(String[] args)
    {
        int x = 13;
        one(x);
        System.out.println(x);
    }

    public static void one(int x)
    {
        x++;
    }
}
```

- A) 0
- B) 13
- C) 14
- D) There is a runtime error at the method call.

X11. What is output when the following code is run?

```
public class Bar
{
    public static void main(String[] args)
    {
        int x = 13;
        x = one(x);
        System.out.println(x);
    }

    public static int one(int x)
    {
        x++;
        return x;
    }
}
```

- A) 0
- B) 13
- C) 14
- D) There is a runtime error at the method call.

X12. What is output when the following code is run?

```
public class Bar
{
    public static void main(String[] args)
    {
        Species x = new Species("Wombat", 200, 2.2);
        one(x);
        System.out.println(x);
    }

    public static void one(Species x)
    {
        x.setPopulation(x.getPopulation() + 1);
    }
}
```

- A) 200
- B) 201
- C) Name: Wombat, Population: 200, Growth Rate: 2.2
- D) Name: Wombat, Population: 201, Growth Rate: 2.2
- E) There is a runtime error at the method call.
- F) There is a runtime error in the one method execution at `getPopulation()`.

X13. What is output when the following code is run?

```
public class Bar
{
    public static void main(String[] args)
    {
        Species x = new Species("Wombat", 200, 2.2);
        x = one(x);
        System.out.println(x);
    }

    public static void one(Species y)
    {
        y.setPopulation = y.getPopulation() + 1;
        y = new Species("Rat", 20, 3.3);
        return y;
    }
}
```

- A) Name: Wombat, Population: 200, Growth Rate: 2.2
- B) Name: Wombat, Population: 201, Growth Rate: 2.2
- C) Name: Rat, Population: 20, Growth Rate: 3.3
- D) Name: Rat, Population: 21, Growth Rate: 3.3

X14. Write a program that will help an instructor find the average grades in their courses. The instructor has 3 sections. They want to find the average grade in each section as well as the average grade over all the sections. The instructor should be prompted to enter the grades (double type scores) for section A00, then for section B00, then for section C00. For each section, the instructor will indicate that they are done entering grades for that section by typing in a negative score.

At the end of entering all the data, the program should output the average for each of the sections and the overall average.

HINT: Make sure you calculate the overall average correctly! If section A00 has an average of 20.0 and both sections B00 and C00 have averages of 90.0, the average MAY NOT BE $(90+90+20)/3$! It depends on how many people are in each section!

X15. It is possible to update the value in a variable declared as `public static final`

True False

X16. It is NOT possible to have a triply nested loop.

True False

X17. Some codes can only be written with for loops – not with while loops.

True False

X18. Use the following nonsensical class to answer the following question

```
public class WierdSpecies
{
    private String name;
    private int population = 0;
    private double growthRate = 1.0;

    public WierdSpecies()
    {
        name = "foo";
        System.out.print("foo");
    }

    public WierdSpecies(String foo)
    {
        name = foo;
        System.out.print(foo);
        this.foo();
    }

    public void foo()
    {
        System.out.print("foo");
    }
}
```

As a result of executing the following two lines of code, how many times will foo be printed?

```
Species foo = new Species("foo");
foo.foo();
```

X19. Write a static method that finds (and returns) the maximum element in an array – up to a certain place (index) in that array. The method takes as parameters an array of double values and an integer which is an index into that array. This index parameter indicates the maximum index we should consider looking at in the array.

X20. Write a static method that finds (and returns) a reference to a `Species` object that has the smallest population of all the objects in an array of `Species`. The method takes as a parameter an array of `Species` objects. You should assume that this method IS NOT written inside the file `Species.java`.