1. (5 points) What are 5 built-in classes you have used?
   i
   ii
   iii
   iv
   v

2. (5 points) Write pseudo-code for the stepwise refinement technique.

3. (2 points) What converts source code to byte code?

4. (2 points) What is the relationship between prototyping and debugging?

5. (5 points) What is the syntax of an assignment statement (for full credit use BNF notation)?

6. (5 points) Write code that will generate a null-pointer Exception.

7. (3 points) `outTF.setText(17);` doesn't compile; why?
   Mark it up so it will.

8. (3 points) A programmer wants to display a FoodItem selected by its name by the user from a ComboBox named foodBox, and writes:
   
   ```java
   outTF.setText(theDB.findByName(foodBox.getSelectedItem()).toString());
   ```
   
   But NetBeans says: method `findByName` in class NewJFrame cannot be applied to given types;
   required: `String`
   found: `Object`
   reason: actual argument `Object` cannot be converted to `String` by method invocation conversion
   
   What's wrong?
   
   Mark the simple change to fix it.
9. (30 points) Write a complete class named Exam with two variables, name and score; write an initializing constructor, the accessors for name and a standard `public String toString()`.
10. (10 points) Write a snippet of code that will output "pass" or "fail" depending on whether the score of an Exam, named myExam, is > 65.

11. (10 points) Write a method isVowel, which returns true if its char parameter is 'a', 'e', 'i', 'o', or 'u'.

12. (5 points) Write a method, named contract, which returns the name of its Exam (see question 9) parameter with all the vowels deleted.

13. (5 points) Declare a class ExamList, which is a synonym for ArrayList<Exam>.

14. (10 points) Write a method, named contract, which will contract the names inside all the Exams in its ExamList parameter.
15. (10 points extra credit) Most natural numbers are the sum of a series of smaller consecutive numbers: e.g. $3=1+2$, $6=1+2+3$, $12=3+4+5$, $17=8+9$. Write a method that for each number from 1-100 will output either "can't do it" or the series of numbers that sums to it. Keep your code simple and understandable (no credit if I cannot understand your code!). No more than 7 lines in any method!