Jeroo Test

Multiple Choice. Circle the best choice.

1. A set of instructions for a computer is called a(n)
   a. algorithm
   b. hardware
   c. program
   d. interpreter

2. The process of creating a specific object is called
   a. instantiation
   b. interpretation
   c. compilation
   d. casting

3. A(n) _____ is a plan for solving a problem.
   a. program
   b. algorithm
   c. recipe
   d. semantics

4. A collection of statements that describe a specific behavior is called a
   a. program
   b. method
   c. syntax
   d. semantics

5. A(n) _____ translates the source code into an intermediate language.
   a. program
   b. algorithm
   c. method
   d. compiler

Code Completion

6. Write a Jeroo method named **turnAround** that instructs a Jeroo to turn around and face the opposite direction.

```
method turnAround()
{
    turn(RIGHT);
    turn(RIGHT);
}
```

7. The main method below has instantiated a Jeroo named beth. Send her a message asking her to perform the **turnAround** behavior defined in question 6. Assume **turnAround** works as specified.

```
method main()
{
   Jeroo beth = new Jeroo();
   beth.turnAround();


}
```

8. Assume there is a Jeroo named beth. There is a row of 10 flowers directly in front of her. Write a while loop that will allow her to pick each of the flowers in the row then stop when the task is complete.

```
while(beth.isFlower(AHEAD))
{
    beth.hop();
    beth.pick();
}
```

9. Assume there is a Jeroo named beth. Write an if/else statement that checks to see if there is water directly in front of her. If there is water in front of her she should turn around and face the opposite direction, otherwise she should hop forward one space. Use the **turnAround** method defined in question 6 in your implementation.

```
if(beth.isWater(AHEAD))
   beth.turnAround();
else
   beth.hop();
```

10. Assume there is a Jeroo named beth. Write an if statement that determines if beth has a flower in her pouch **and** there is a net directly in front of her. If this condition is true beth needs to toss the flower on to the net to disable it and hop forward one space to occupy the space left by the net.

```
if(beth.hasFlower() && beth.isNet(AHEAD))
{
   beth.toss();
   beth.hop();
}
```