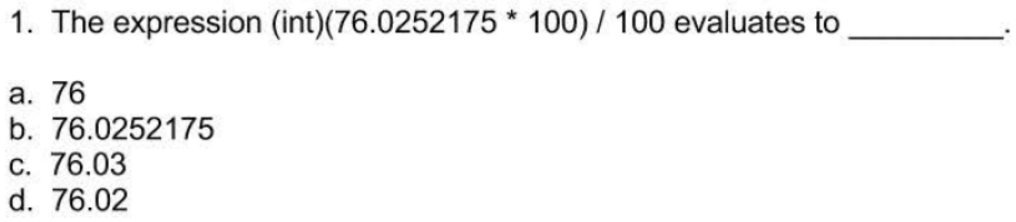
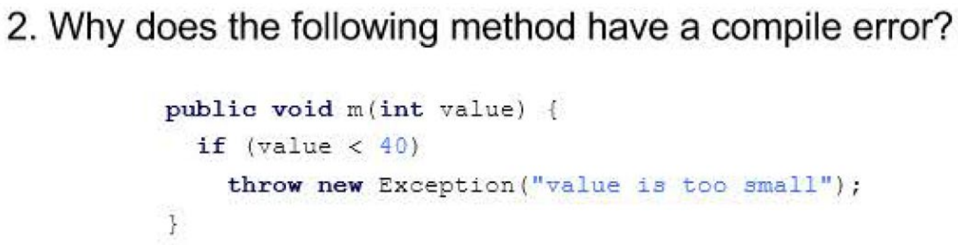
Java Exam



1. The expression (int)(76.0252175 * 100) / 100 evaluates to _________.

a. 76
b. 76.0252175
c. 76.03
d. 76.02

a.) 76



2. Why does the following method have a compile error?

```
public void m(int value) {
  if (value < 40)
    throw new Exception("value is too small");
}
```

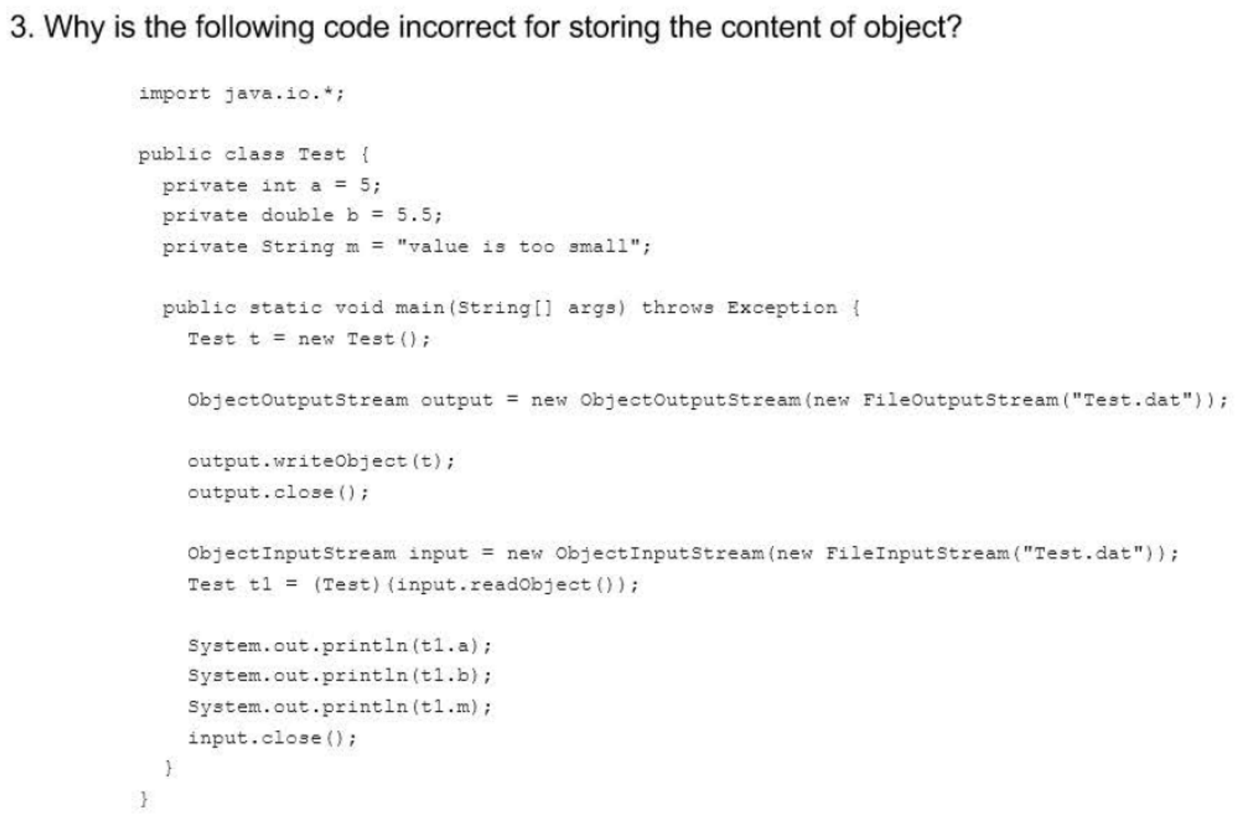
The method doesn’t require throwing an Exception. You’re already using an if clause. A fix for this method would just be replacing the Exception with a print statement:

System.out.println(“value is too small”);

You don’t throw an exception inside an if clause like this. As for the specific compile error, the exception is unhandled. The fix would be:

public void m(int value) throws Exception {

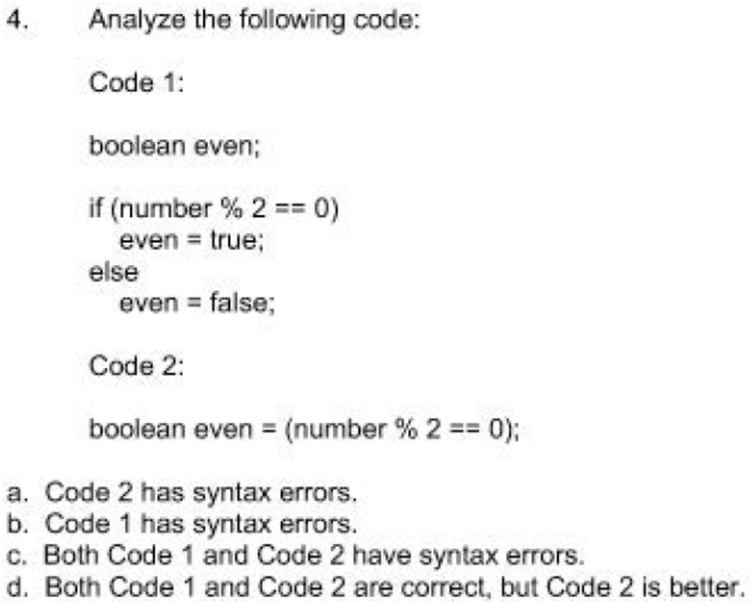
For the first line



3. Why is the following code incorrect for storing the content of object?

```
import java.io.*;

public class Test {
  private int a = 5;
  private double b = 5.5;
  private String m = "value is too small";

  public static void main(String[] args) throws Exception {
    Test t = new Test();

    ObjectOutputStream output = new ObjectOutputStream(new FileOutputStream("Test.dat"));

    output.writeObject(t);
    output.close();

    ObjectInputStream input = new ObjectInputStream(new FileInputStream("Test.dat"));
    Test t1 = (Test)(input.readObject());

    System.out.println(t1.a);
    System.out.println(t1.b);
    System.out.println(t1.m);
    input.close();
  }
}
```

The input reads an object of Type Object and casts it to type Test in order to store it in object t1. That is the incorrect way to do this. The program also never properly initialized object t1.



4. Analyze the following code:

Code 1:

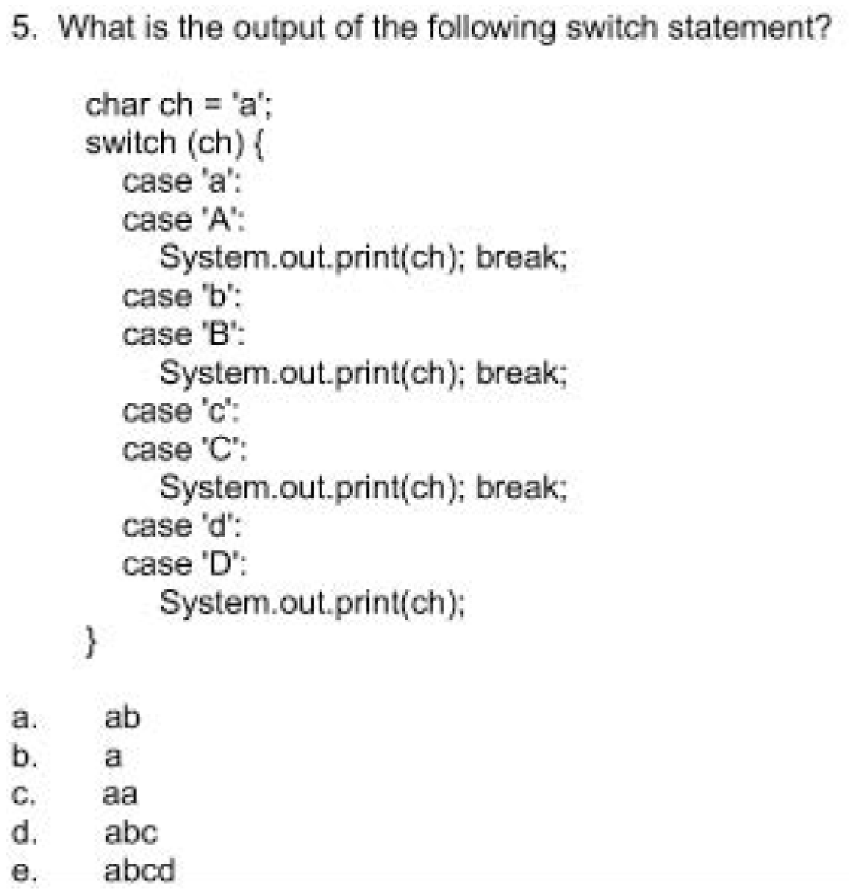
```
boolean even;

if (number % 2 == 0)
   even = true;
else
   even = false;
```

Code 2:

```
boolean even = (number % 2 == 0);
```

a. Code 2 has syntax errors.
b. Code 1 has syntax errors.
c. Both Code 1 and Code 2 have syntax errors.
d. Both Code 1 and Code 2 are correct, but Code 2 is better.

C.) Both Code 1 and Code 2 have syntax errors: number was never initialized

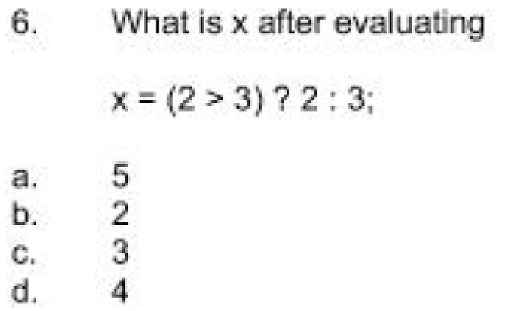


5. What is the output of the following switch statement?

```
char ch = 'a';
switch (ch) {
   case 'a':
   case 'A':
      System.out.print(ch); break;
   case 'b':
   case 'B':
      System.out.print(ch); break;
   case 'c':
   case 'C':
      System.out.print(ch); break;
   case 'd':
   case 'D':
      System.out.print(ch);
}
```

a. ab
b. a
c. aa
d. abc
e. abcd

b. a

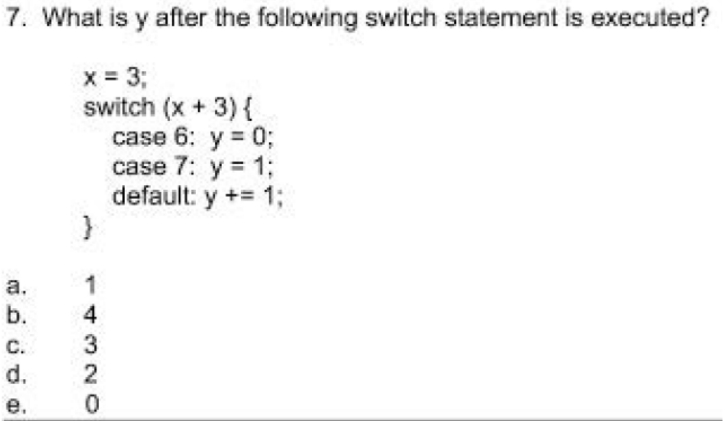


6. What is x after evaluating

x = (2 > 3) ? 2 : 3;

a. 5
b. 2
c. 3
d. 4

c.) 3

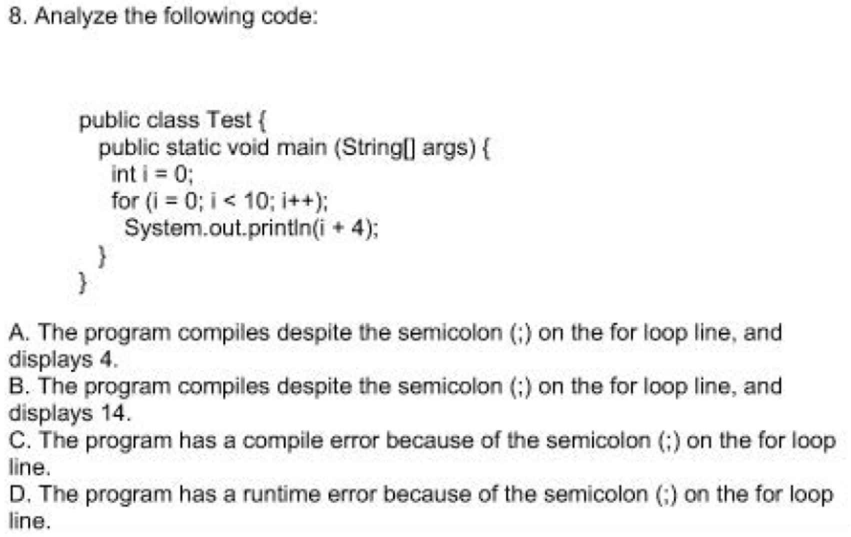


7. What is y after the following switch statement is executed?

```
x = 3;
switch (x + 3) {
   case 6:  y = 0;
   case 7:  y = 1;
   default: y += 1;
}
```

a. 1
b. 4
c. 3
d. 2
e. 0

d.) 2

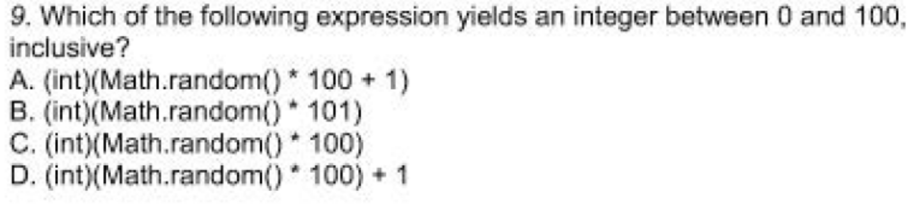


8. Analyze the following code:

```
public class Test {
  public static void main (String[] args) {
    int i = 0;
    for (i = 0; i < 10; i++);
      System.out.println(i + 4);
  }
}
```

A. The program compiles despite the semicolon (;) on the for loop line, and displays 4.
B. The program compiles despite the semicolon (;) on the for loop line, and displays 14.
C. The program has a compile error because of the semicolon (;) on the for loop line.
D. The program has a runtime error because of the semicolon (;) on the for loop line.

A.) The program compiles despite the semicolon (;) on the for loop line, and displays 4.

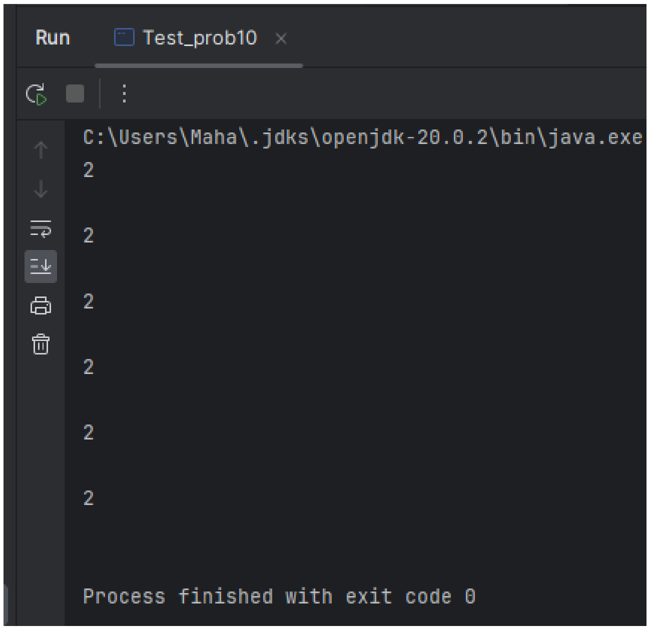


9. Which of the following expression yields an integer between 0 and 100, inclusive?
A. (int)(Math.random() * 100 + 1)
B. (int)(Math.random() * 101)
C. (int)(Math.random() * 100)
D. (int)(Math.random() * 100) + 1

B.) (int) (Math.random() * 101)

10. Show the output of the following code:

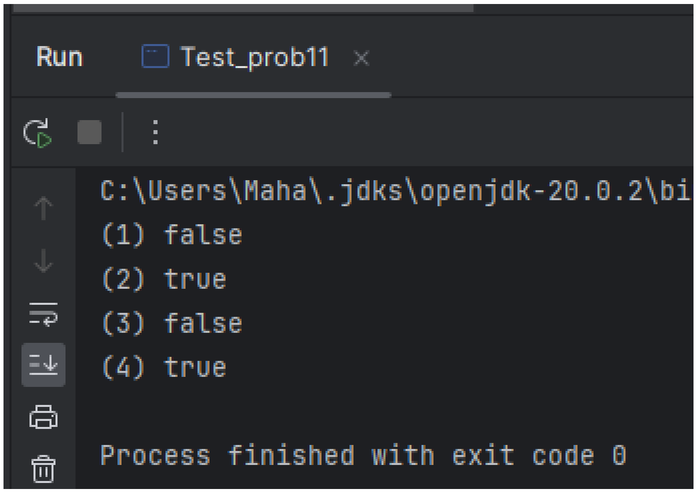
```
public class Test {
  public static void main(String[] args) {
    int i = 1;
    while (i <= 6) {
      method1(i, 2);
      i++;
    }
  }

  public static void method1(int i, int num) {
    for (int j = 1; j <= i; j++) {
      System.out.print(num + " ");
      num *= 2;
    }

    System.out.println();
  }
}
```



```
C:\Users\Maha\.jdks\openjdk-20.0.2\bin\java.exe
2

2

2

2

2

2


Process finished with exit code 0
```

11. The java.util.Date class overrides the equals method to return true if two objects have the same date and time. Show the output of the following code.

```
import java.util.*;
public class Test extends Object {
  public static void main(String[] args) {
    Date d1 = new Date();
    Date d2 = new Date(349324);
    Date d3 = d1;
    System.out.println("(1) " + (d1 == d2));
    System.out.println("(2) " + (d1 == d3));
    System.out.println("(3) " + d1.equals(d2));
    System.out.println("(4) " + d1.equals(d3));
  }
}
```



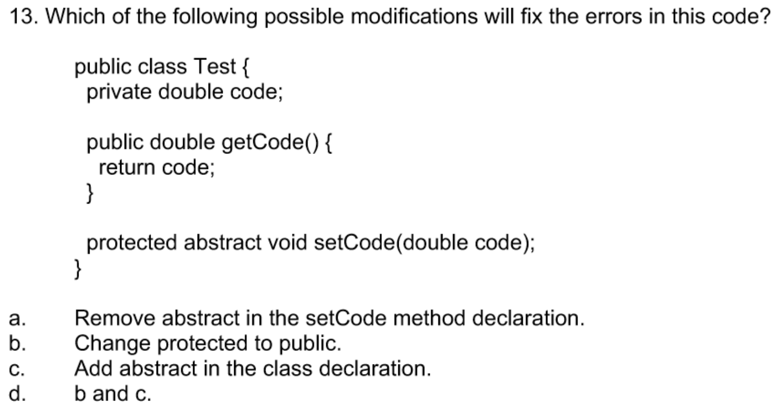
```
C:\Users\Maha\.jdks\openjdk-20.0.2\bi
(1) false
(2) true
(3) false
(4) true

Process finished with exit code 0
```

12. What exception type does the following program throw?

```
public class Test {
   public static void main(String[] args) {
      Object o = new Object();
      String d = (String)o;
   }
}
```

a. ArithmeticException
b. No exception
c. StringIndexOutOfBoundsException
d. ArrayIndexOutOfBoundsException
e. ClassCastException

e.) ClassCastException

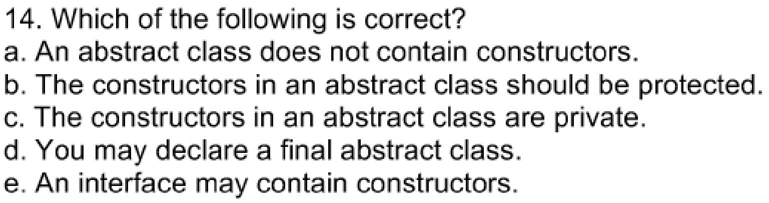


13. Which of the following possible modifications will fix the errors in this code?

```
public class Test {
  private double code;

  public double getCode() {
    return code;
  }

  protected abstract void setCode(double code);
}
```

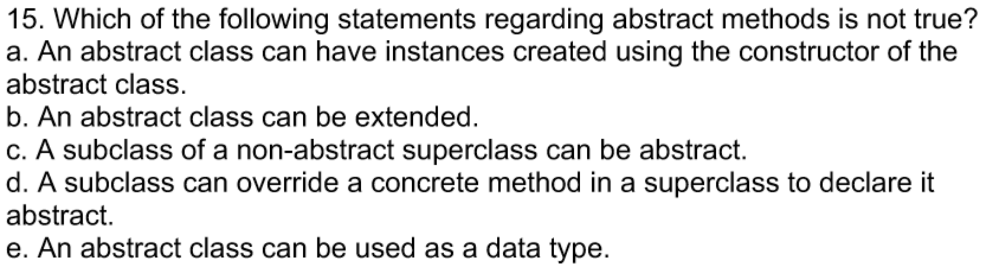
a. Remove abstract in the setCode method declaration.
b. Change protected to public.
c. Add abstract in the class declaration.
d. b and c.

b.) Add abstract in the class declaration



14. Which of the following is correct?
a. An abstract class does not contain constructors.
b. The constructors in an abstract class should be protected.
c. The constructors in an abstract class are private.
d. You may declare a final abstract class.
e. An interface may contain constructors.

B.) The constructors in an abstract class should be protected



15. Which of the following statements regarding abstract methods is not true?
a. An abstract class can have instances created using the constructor of the abstract class.
b. An abstract class can be extended.
c. A subclass of a non-abstract superclass can be abstract.
d. A subclass can override a concrete method in a superclass to declare it abstract.
e. An abstract class can be used as a data type.

A.) An abstract class can have instances created using the constructor of the abstract class

Short Answers

16. What is the difference between Pointers in C and Reference Variables in Java?

Arithmetic can be performed on C pointers but not on Java reference variables.

17. Explain in your own words what is the difference between overriding and overloading?

Overriding changes the implementation of a method already defined in a parent method. Overloading is when you allow different variable types for the same method name (i.e. max (int x, int, y) and max(doube x, double y). The method signature is changed for overloading because of the change in parameters.

18. What is a static method, when should we use it?

A static method is a method that belongs to the class. Unlike an instance method that belongs to the object. You should use a static method when the method and the variables related to the method don’t change from object to object, but remain the same across all objects as long as the class is the same.

19. What is a nested class? with an example explain what a good use case would be.

A nested class is when a class is defined within another. A good example is when defining your own Linked List class. You can create a Node class within the Linked List class.

20. What is primitive data type? Give an example of when a primitive gets stored in the heap memory.

A primitive data type is one that is provided by default with the language. For example, Java automatically provides int, double and boolean as primitive data types. A primitive data type can get stored in the heap as instance variables.

21. Explain in your own words what is dynamic typing?

Dynamic typing is when the compiler decides what type a variable will be when the program is run. Python is an example of a programming language that uses dynamic typing because unlike Java, you don’t declare the type of the variable. It’s decided at run time depending on the value of the variable.

22.With some examples in your own words explain what an object of a class in OOP paradigm is?

An object in the OOP paradigm is meant to be abstract and kind of mimic an object in real life. Data fields are meant to mirror attributes of the objects. And methods are meant to mirror the actions the object can partake in. For example, you can create an animal object and give it attributes of colour, height, weight etc, and give it actions

such as eat, run and sleep. And you can apply Inheritance to it and create Mammal objects under the animal object and feline and canine objects under the animal object and so on and so forth.

23. Explain Stackoverflow error.

A stack overflow error is when the stack tries to use more memory space (determined by software and hardware limitations) than it has access to. Stack overflow errors usually result in programs crashing.

24. What does memory allocation for an object mean?

For an object to exist and be created, it needs a certain amount of memory to be able to take up space (a memory address). Further memory is thus needed to store its member variables and functions.