THE UNIVERSITY OF WESTERN AUSTRALIA

MID-SEMESTER EXAMINATION
SEPTEMBER 2011

SCHOOL OF COMPUTER SCIENCE & SOFTWARE ENGINEERING

JAVA PROGRAMMING CITS1200

This Paper Contains:
15 Pages
15 Questions

Time allowed: FIFTY MINUTES

Marks for this paper total 15.
Candidates should answer ALL Questions.
Q1. Variables of which of these types cannot be added using the + operator?

a. int
b. double
*c. boolean
d. char
e. String

Q2. Consider the following method (with line numbers):

1. public boolean justAMethod(int i, int j, k) {
2.     boolean n = false;
3.     int m = 0
4.     if (i > j)  
5.         return true;
6.     return n;
7. }

There are syntax errors in:

a. line 3
b. lines 1, 2 and 3
c. lines 1, 2 and 5
*d. lines 1 and 3
e. lines 1 and 2
Q3. Consider the following method:

```java
class MysteryMethod {
    public int mystery(int j) {
        int result = 1;
        if (j == 1) {
            return j;
        }
        for (int i = 1; i <= j; i++) {
            result = result * i;
        }
        return result;
    }
}
```

What does `mystery(5)` return?

a. 1  
b. 120  
c. 15  
d. 12  
e. 0  

Q4. How many iterations does this loop perform?

```java
for (int i = 1; i < 10; i *= 2) {
    i++;
}
```

*a. 2  
b. 3  
c. 4  
d. 5  
e. 10  

Q5. Consider the following method:

```java
public int anotherMystery(int i, int j){
    int k = 0;
    k = i;
    i = j;
    j = k;
    return j;
}
```

What does `anotherMystery(10,15)` return?

a. 15  
b. 10  
c. 25  
d. 5  
e. 0
Q6. What is the value of \texttt{method(8)}, where \texttt{method} is defined as follows?

\begin{verbatim}
public int method(int n) {
    if (n == 1 || n == 2)
        return 1;
    int last = 1;
    int current = 1;
    int next;
    for (int i=3; i<=n; i++) {
        next = current + last;
        last = current;
        current = next;
    }
    return current;
}
\end{verbatim}

a. 2
b. 8
c. 13
*d. 21
e. 55
Q7. How many of these statements about constructors are true?

- All constructors in a class must have different return types.
- All constructors in a class must have different argument types.
- Every class must have at least one constructor.
- Constructors cannot initialise class variables.

a. 0
*b. 1

c. 2

d. 3

e. 4

Q8. What does secondMystery(100) return?

```java
public int secondMystery(int k) {
    while (k > 10) {
        k = k / 2;
    }
    return k;
}
```

a. 2
*b. 6

c. 100

d. 12

e. 5
Q9. Which of these diagrams most closely resembles the picture drawn by `drawPicture`?

```java
public void drawPicture()
{
    SimpleCanvas c = new SimpleCanvas();
    for (int i=0; i<=400; i=i+80)
        c.drawLine(i, 399, 399, i);
}
```

a. ![Diagram A]

b. ![Diagram B]

c. ![Diagram C]

d. ![Diagram D]

e. ![Diagram E]
Q10. What does booleanMystery(false,true) return?

```java
public int booleanMystery(boolean b1, boolean b2){
    if ((b1 && b2) == true)
        return 100;
    else if ((b1 || b2) == true)
        return 200;
    else return 300;
}
```

a. 100
*b. 200*

c. 300
d. true
e. false

Q11. How many of these statements apply to a large program written in good object-oriented style?

- The problem is decomposed into several classes.
- Each class provides a narrow range of well-defined services.
- Each class hides its implementation details as far as possible.
- Objects communicate as little as possible at runtime.

a. 0
b. 1
c. 2
*d. 3*
e. 4
Q12. Consider the following class definition

```java
public class BankAccount {
    private int balance;
    private static int total = 0;

    public BankAccount(int balance) {
        this.balance = balance;
        total = total + balance;
    }

    public void deposit(int amount) {
        balance = balance + amount;
        total = total + amount;
    }

    public void withdraw(int amount) {
        balance = balance - amount;
        total = total + amount;
    }

    public int getBalance() {
        return balance;
    }

    public int getTotal() {
        return total;
    }
}
```
What would \texttt{b1.getTotal()} and \texttt{b2.getTotal()} return after these statements?

```java
BankAccount b1 = new BankAccount(1000);
BankAccount b2 = new BankAccount(500);
b1.deposit(1500);
b2.withdraw(200);
b1.deposit(b2.getBalance());
```

a. 1700 and 1700  

b. 2500 and 1300  

c. 2800 and 300  

d. 3100 and 3100  

*e. 3500 and 3500
Questions 13, 14 and 15 are based on the following two classes:

```java
public class First{
    private int anInt;
    private Second second;
    public First(){
        this.anInt = 10;
        this.second = new Second();
    }

    public int returnAnInt(){
        return anInt;
    }

    public int returnOtherInt(){
        return this.second.returnAnInt();
    }
}

public class Second{
    private int anInt;
    private First first;

    public int returnAnInt(){
        return anInt;
    }

    public int returnOtherInt(){
        this.anInt = 20;
        this.first = new First();
        return (this.returnAnInt()+this.first.returnAnInt());
    }
}
```
Q13. If we create an instance of Second and execute the method returnOtherInt() of this instance, the result will be:

a. 20
*b. 30

c. 10
d. 15
e. 0

Q14. If you write the following method in class Second, compile the class and execute the method:

```java
public int compareAndReturn(){
   this.first = new First();
   if (this.returnAnInt() >= this.first.returnAnInt())
      return 12;
   else return 15;
}
```

You will get a return value:

*a. 15
b. 30
c. 10
d. 12
e. 20
Q15. If you write the following method in class Second, compile the class and execute the method:

```java
public int compareAndReturn2(){
    this.first = new First();
    if (this.first.returnAnInt() >= this.returnAnInt())
        return 15;
    else return 12;
}
```

You will get a return value:

*a. 15
b. 30
c. 10
d. 12
e. 20*
This page is for rough work.
This page is for rough work.