King Saud University
College of Computer and Information Sciences
Computer Science Department

CSC 111 — First Semester
Introduction to Programming with Java — 1440-1441

Enjoy :)

Java Final Mock Exam

23-12-2019


-جسّوم

Question	Points	Student Score	Total "divide by 2"
True or False	20		
Multiple Choice	20		
Tracing	10		
Errors	10		
Code Segment	10		
Program	10		

Q#1 True or False:

Constructor overloading is not possible in Java.	**F**
consider the statement "x = (a < b) ? a : b"; then the value of x is 27, if a = 27 and b = 18	**F**
An array in the Java programming language has the ability to store many different types of values	**F**
A *static* method cannot refer to any instance variable of the class	**T**
Methods can be overloaded with a difference only in the type of the return variable.	**F**
"int num = 8/0" will result in Compilation error: DivideByZeroException	**F**
The concept of abstraction is so that the programmer using a class method does not need to know the details of implementation	**T**
When calling a Java method the programmer required to explicitly provide the type information for each parameter	**F**
A class attribute is in one fixed location in memory, meaning all objects can access it	**T**
Given a class with public attribute Att, the following code is invalid: ObjectName.Att = TypeValue;	**F**
Accessor methods are designed to get information about an object	**T**
The main has its own method invocation	**T**
A constructor can initialize and return values	**F**
A method can have multiple return statements	**T**
Ann array cannot have a length 0	**F**
Given an arbitrary array A1 and another array A2, this expression will always be false: A1==A2. As it compares addresses.	**F**
Its invalid to include [] in the actual parameter of a method	**T**
This code will cause an infinite loop: *for*(;;) *if*(1 > 2) System.out.println(); *else* System.out.println();	**T**
This code will cause a syntax error: System.out.print();	**T**
int arr[] = new int[3] {1,2,3}; **is a valid statement**	**F**

Q#2 MCQ:

A _____ is a program that executes compiled Java code on a specific platform.	**a) Java Virtual Machine** **b) Java Compiler** **c) Java Programming Manual** **d) Eclipse Editor** **e) None of the above**
Which of the following is not a Java keyword?	**a) public** **b) for** **c) input** **d) static** **e) None of the above**
Following code will result in: int a1 = 5; double a2 = (float)a1;	**a) Compile error** **b) Run-time error** **c) Out of bound exception** **d) Type casting exception** **e) None of the above**
Which of following declarations is valid?	**a) long a, b, a;** **b) float x, int;** **c) byte x, y = 13;** **d) double x, long y;** **e) None of the above**
Suppose we have the following declarations with arbitrary values: int i, j; float x, y; double a, b; **Which of following assignment is invalid?**	**a) i = b+j;** **b) i = (*int*)b/j*i;** **c) y = j / i * x;** **d) b = i*j*x/y%i;** **e) None of the above**

What is the output of the following code:
int x = 1, y = 2;
do{
System.out.print("JAVA");
}
while (x < y)
System.out.print("CSC111");
	a) Infinite loop
	b) Compilation error
	c) CSC111
	d) Run time error
	e) None of the above

What is the output of the following code: int sum; for(sum=0; sum>=0; sum++) sum--; System.out.println("sum: " + sum);	**a) 0** **b) -1** **c) Infinite loop** **d) Compilation error** **e) None of the above**

How many loops will iterate:
int n=50;
while (n>=10)
n-=n/n;
	a) 10
	b) 25
	c) 40
	d) 50
	e) None of the above

At the end of the run the value of num would be:
int num = 1;
if(num++ == num);
num += num;
else
num = 23;
System.out.println(num);
	a) 4
	b) 23
	c) 1
	d) Compilation error
	e) None of the above

The output of the following code is :
int OldArr[] = {1,1,2,0};
int NewArr[] = {1,2,-1,23};
for(int i = 0; i < 4; i++)
NewArr[OldArr[i]] = NewArr[i];
for(int i = 0; i > 4; i++)
System.out.print(NewArr[i]);
	a) 231-123
	b) Nothing
	c) 12-123
	d) 1120
	e) None of the above

Q#3 Trace the following code:

```
public class PrimeNum
{
private int n;
PrimeNum ()
  {
n = 2;
}

void setPrimeNumber (int
num)
  {
n = num;
}

int getPrimeNumber ()
  {
return n;
}

boolean isPrimeNumber ()
  {
for (int i = 2; i < n; i++)
     {
if (n % i == 0)
              {
return false;
}
}
return true;
}
}
```

```
import java.util.Scanner;

public class PrimeNumTest
{
static Scanner input = new Scanner (System.in);
public static void main (String[]args)
  {
PrimeNum prime1 = new PrimeNum ();
int num = 7;
prime1.setPrimeNumber (num);
if (prime1.isPrimeNumber ())
     {
System.out.println ("The number you entered is " + num" and the
prime numbers less than " + num + " are:");
PrimeNum[]p = new PrimeNum[num];
for (int i = 0; i < num; i++)
              {
p[i] = new PrimeNum ();
p[i].setPrimeNumber (i + 2);
}
for (int j = 0; j < num; j++)
              {
if (p[j].isPrimeNumber () && p[j].getPrimeNumber () != num)
System.out.println (p[j].getPrimeNumber ());
}
}
   else
System.out.println ("The number you entered is not prime");
}
}
```

Output:

The number you entered is 7 and the prime numbers less than 7 are:

2

3

5

Q#4 Find the errors in the following program:

```
public class FindErrorsClass{

private int att1;
private String att2;
private double att3;
int count = 0;

FindErrorsClass(){
att1 = 0;
att2 = null;
att3 = 0;
count++;
return true; *
}
int getAtt1(){
return att1;
}
private void setAtt2(String a){
att2 = a;
}
void setAtt3(double d){
att3 = d;
}
double calculateValue(){
return att3+att1;
}
}
```

*Cannot have a return value in a constructor

*Constructor does not take any parameters

*Cannot access private attributes

*Must write object name

*There is not a method with this name

```
import java.util.Scanner;
public class test{
static final int arrsize = 2;
public static void main(String[] args){
Scanner input = new Scanner(System.in);
FindErrorsClass arr [] = new FindErrorsClass[arrsize];
FindErrorsClass obj1 = new FindErrorsClass(22,3.2,":)");*
FindErrorsClass obj2 = new FindErrorsClass();
obj2.count = 2;
obj2.setAtt3(22.2);
obj2.att3 = 12.5;*
double elements = calculateValue();*
for(int i = 0; i < 2; i++){
arr[i] = new FindErrorClass();
arr[i].setAtt2 = ":/"; *
}
}
}
```

Q#5 Write a code segment:

1. Given a method header shiftElements(char OldArr[], char c) write the body which replaces the given character in the array with zero and shifts them to the end of the array.

How the output should look:

The given array → {a,v,a,%,n}

The output → {v,%,n,0,0}

```
static char[] shiftElements(char[] OldArr, char c){

char [] NewArr = new char[OldArr.length];//a new array to add the values and return
char TempVar = 0;

if(OldArr[OldArr.length - 1]!= c)
TempVar = OldArr[OldArr.length -1];//saving the last char in a temporary value since it wont get copied

int j = 0;//counter for the new array indices

for(int i = 0; i < OldArr.length-1; i++){
if(OldArr[i] == c)//skips if the array has the character c
continue;
NewArr[j] = OldArr[i];
j++;
}

NewArr[j] = TempVar;//adding the variable in the index j

for(int i = 0; i < OldArr.length; i++)//filling the null values with 0
if(NewArr[i] == 0)
NewArr[i] = '0';
return NewArr;
```

2. Write a method that accepts a password and returns true if the password is valid, a valid password is when the following is checked:

The password has at least a length of 8

The password has at least a capital letter

The password has at least a digit

Consider the method header checkPassword(String str).

```
static boolean checkPassword(String str){
boolean checkValid = false;
int j = 0, k = 0;
for(int i = 0; i < str.length(); i++){
if(Character.isDigit(str.charAt(i)))
j++;
if(Character.isUpperCase(str.charAt(i)))
k++;
}
if(str.length() >= 8 && j >= 1 && k >= 1 )
checkValid = true;
return checkValid;
}
```

3. Given the following declaration → int[] arrFindSum = {6,1,2,3,0,0}; write a code segment that compares the total sum of the array to the first element.

```
int sum = 0;
for(int i = 0; i < arrFindSum.length; i++)
sum += arrFindSum[i];

System.out.println(arrFindSum[0]==sum);
```

Q#6 Write a program:

Consider a class Time that represents a time of day. It has attributes for the hour and minute. The hour value ranges from 0 to 23, where the range 0 to 11 represents a time before noon. The minute value ranges from 0 to 59.

Time
- hour: int **- min: int**
+Time(int x, int y): **-isValid(int x, int y): boolean** **+setTime(int x, int y): void** **+setTime(int x , int y, String isAM): void** **+displayTime():void**

the default constructor initializes the time to 0 hours, 0 minutes.
the method isValid(hour, minute) returns true if the given hour and minute values are in the appropriate range.
the method setTime(hour, minute) that sets the time if the given values are valid.
write another method setTime(hour, minute, isAM) that sets the time if the given values are valid. The given hour should be in the range 1 to 12. The parameter isAm is true if the time is an a.m. time and false otherwise.
The method displayTime(hour, minute) simply dispalys the time in the format HH:MM
Hint: The second version of setTime() (an overloaded method) must check to make sure that hour is <= 12, and should convert a p.m. hour (one where isAm is false) to that hour + 12 to convert it to a correct afternoon time; in either case, it should then just call the first version of setTime().

```
public class Time{

private int hour;
private int min;

Time(){
hour = 0;
min = 0;
}

private boolean isValid(int x, int y){
if(x >= 0 && x <= 23 && y >= 0 && y <= 59)
return true;
return false;
}

void setTime(int x, int y){
if(isValid(x,y)){
hour = x;
min = y;
}
}
```

```
void setTime(int x, int y, boolean isAM){
if(x >=1 && x <= 12 && isAM )
setTime(x,y);
if(!isAM && x >= 13 && x < 24){
x = x-12;
setTime(x,y);
}
}

void displayTime(){
System.out.println(hour+":"+min);
}
}
```

Write a main method that instantiates a Time object and invokes setTime more than one waya and displays the output.

```
public static void main(String[] args){

Time t1 = new Time();
t1.setTime(12,55);
t1.setTime(24,55,false);
t1.displayTime();
}
```