

## Homework/Program #1 Solutions

1.

a) ( 1 point for each Hello\*.java files (3 points), 1 point for each Hello\*.class file (3 points)

```
-rw-r--r--. 1 cs11xyz cs11sxyz 718 Oct 2 21:47 Hello2.class
-rw-r--r--. 1 cs11xyz cs11sxyz 938 Oct 2 21:22 Hello2.java
-rw-r--r--. 1 cs11xyz cs11sxyz 427 Oct 2 21:47 HelloWorld.class
-rw-r--r--. 1 cs11xyz cs11sxyz 610 Oct 2 21:47 HelloWorldGraphics.class
-rw-r--r--. 1 cs11xyz cs11sxyz 882 Oct 2 21:14 HelloWorldGraphics.java
-rw-r--r--. 1 cs11xyz cs11sxyz 477 Oct 2 21:13 HelloWorld.java
-rw-r--r--. 1 cs11xyz cs11sxyz 130 Oct 2 21:14 myapplet.html
-rw-r--r--. 1 cs11xyz cs11sxyz 67068 Oct 2 21:23 objectdraw.jar
-rw-r--r--. 1 cs11xyz cs11sxyz 0 Oct 2 22:16 PR1.txt
```

b) (4 points total)

Application: ( 2 points for two of the three following)

- 1) Applications have main method
- 2) Applications are executed as standalone programs
- 3) Applications do not have access restrictions

Applet: ( 2 points for two of the three following)

- 1) Applets do not have a main method
- 2) Applets require some third party tool help like a browser or appletviewer to execute
- 3) Applets have restricted access to local computer (e.g. no access to local file system)

2.

(2.5 x 4=10)

(1 point for error, 1.5 point for each explanation in plain English)

The following are examples of errors. Others are possible and valid

```
public class HelloWorld
{
    public static void main(String[] args)
    {
        System.out.println("Hello, World!");
    }
}
```

a.1) void public static main(String[] args)-- Wrong

a.2) public static void main(String[] args) – Correct

a.3) HelloWorld.java:14: <identifier> expected

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

^

HelloWorld.java:14: '(' expected

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

^

HelloWorld.java:14: invalid method declaration; return type required

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

a.4) return type should be before the function

b.1) `system.out.println("Hello, World!");` -- Wrong

b.2) `System.out.println("Hello, World!");` -- Correct

b.3) HelloWorld.java:16: package system does not exist

```
system.out.println("Hello, World!");
```

b.4) S has to be capital

c.1) `System.out.println("Hello, World!")` -- Wrong

c.2) `System.out.println("Hello, World!");` -- Correct

c.3) HelloWorld.java:16: ';' expected

```
system.out.println("Hello, World!")
```

^

c.4) Semi colon is missing

d.1) `System.out.println('Hello, World!);` -- Wrong

d.2) `System.out.println("Hello, World!");` -- Correct

d.3) HelloWorld.java:16: unclosed character literal

```
System.out.println('Hello, World!);
```

^

HelloWorld.java:16: ';' expected

```
System.out.println('Hello, World!);
```

^

HelloWorld.java:16: illegal start of expression

```
System.out.println('Hello, World!);
```

^

HelloWorld.java:16: ';' expected

```
System.out.println('Hello, World!');
```

^

HelloWorld.java:16: unclosed character literal

```
System.out.println('Hello, World!');
```

d.4) double quotes used instead of single quotes for string

3. **(1 point for each correct answer + 1 point for attempting problem = 5 points )**

class – class is a keyword used to define a class

Hangman – Hangman is name of the class

extends -extends is a keyword used to inherit properties of another class

WindowsController- Hangman class inherits properties from WindowsController class

4. **(1 point for each correct answer)**

**(1 x 10= 10)**

- a) Valid
- b) Not Valid
- c) Valid
- d) Valid
- e) Valid
- f) Not Valid
- g) Valid
- h) Valid
- i) Not Valid
- j) Not Valid

5. **(1 point for each correct answer)**

**(1 x 5 =5)**

- a – (0,0,0)
- b – (255,255,255)
- c -- (160, 32, 240)
- d – (255,255,0) or (128,128,0)
- e – (0,0,128)