

Java – Programming Examples

- Here are some Java examples of concepts you learned in ICS3U using C#
- Use this as a reference, as you move forward with the more complex ICS4U projects

The main method includes the statements that are run when the program is executed. The starting place of your program

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

}
```

Display text to the screen (note the difference between print() and println())

```
System.out.print("Grade 12");
System.out.println("Computer Science");
System.out.println("St. Paul High School ");
```

Output:

Grade 12Computer Science
St. Paul High School

Display a mix of strings and variables:

```
int birthMonth = 5;
int birthday = 28;
System.out.println("Your birthday is " + birthMonth + "/" + birthday);
```

Prompt the user for a number

```
int playerNum;
Scanner input = new Scanner(System.in);
System.out.print("Enter your number: ");
playerNum = input.nextInt();
input.close();
```

Generate a random integer between 1 and 3

```
int computerThrow = (int)(3 * Math.random() + 1);
```

Switch Statement

```
final int ROCK = 1, PAPER = 2, SCISSORS = 3;
int playerThrow = 3;

switch (playerThrow) {
    case ROCK: System.out.println("ROCK."); break;
    case PAPER: System.out.println("PAPER."); break;
    case SCISSORS: System.out.println("SCISSORS."); break;
}
```

Output:

SCIZZORS

if and else Statements

```
final int ROCK = 1, PAPER = 2, SCISSORS = 3;
int playerThrow = 1; //ROCK
int computerThrow = 3; //SCIZZORS
if (playerThrow == ROCK && computerThrow == ROCK) {
    System.out.println("It's a draw!");
} else if (playerThrow == ROCK && computerThrow == PAPER) {
    System.out.println("Computer wins!");
} else if (playerThrow == ROCK && computerThrow == SCISSORS) {
    System.out.println("Player wins!");
}
```

Output:

Player wins!

ICS4U Module 3: Note ↓ Exercise 1a

do while statement

```
do {  
    System.out.print("Enter a number less than 4:");  
    playerNum = input.nextInt();  
  
} while (playerNum >= 4);
```

this do-while loop
prompts the user
until a valid number
is entered.

while statement

```
int num = 0;  
while (num < 5) {  
    num += 1;
```

This loop
executes 5
times

After the 5th
execution, num is
equal to 5 - making
the condition false

for loop

```
for (int i = 1; i <= 10; i++) {  
    System.out.println(i);  
}
```

since i is declared in the
initialization, the scope of
this counter is only within
this loop - this is good
programming style!

Output:

1
2
3
4
5
6
7
8
9
10

Declaring an array

declares an array of strings called friends to hold 5 strings

```
String[] friends = new String[5]
```

OR

declares an array of strings called friends to hold as many strings as the user wants

```
numFriends = input.nextInt();
```

```
String[] friends = new String[numFriends];
```

OR

declares an array of strings called friends to hold these 5 strings

```
String[] friends = {"Kermit", "Lucille", "Sammy", "Roxy", "Myah"};
```

Using an Array

```
System.out.println(friends[2]);
```

displays Sammy

```
friends[2] = "Sunshine";
```

changes Sammy to Sunshine

```
friends[5] = "McDougall";
```

ERROR — past bounds of the array

```
numElements = friends.length();
```

numElements will be 5

```
for (int i = 0; i < friends.length(); i++) {  
    System.out.println(friends[i]);  
}
```

loops through array to display all names