REFERENCES, POINTERS AND STRUCTS

Problem Solving with Computers-I

Modify the function to swap the values of a and b: use pointers

```
void swapValue(int x, int y){
    int tmp = x;
    x = y;
    y = tmp;
}

int main() {
    int a=30, b=40;
    swapValue( a,  b);
    cout<<a<<" "<<b<<endl;
}
```

Draw the pointer diagram for your code

Segmentation faults (aka segfault)

- Segfault: your program has crashed!
- What caused the crash?
  - Read or write to a memory location that either doesn't exist or you don't have permission to access
  - Dereferencing a null pointer

- Avoid segfaults in your code by
  - Always initializing a pointer to null upon declaration
  - Performing a null check before dereferencing it
  - Avoid redundant null checks by specifying pre and post conditions for functions that use pointers

```
int *p;
*p = 5;
```

Q: Which of the following is true about the above code?

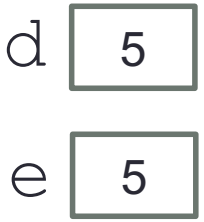
A	Compile time error
B	Runtime error
C	Code runs without error

References in C++

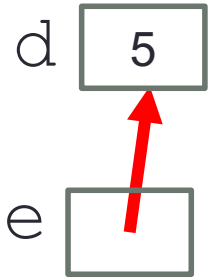
```
int main() {
   int d = 5;
   int &e = d;
}
```

A reference in C++ is an alias for another variable

A.



B.



C.

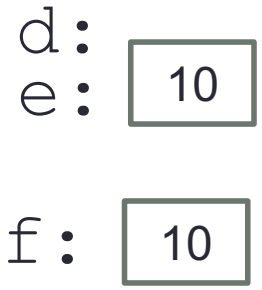
D. This code causes an error

References in C++

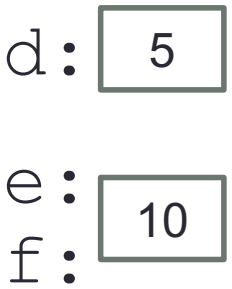
```
int main() {
   int d = 5;
   int & e = d;
   int f = 10;
   e = f;

}
```

How does the diagram change with this code?

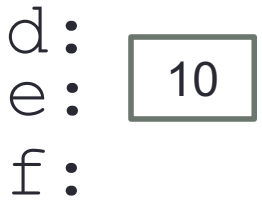
A.



B.



C.



D. Other or error

Pointers and references: Draw the diagram for this code

```
int a = 5;
int & b = a;
int* pt1 = &a;
```

What are three ways to change the value of ‘a’ to 42?

Call by reference: Modify to correctly swap a and b

```
void swapValue(int x, int y){
    int tmp = x;
    x = y;
    y = tmp;
}

int main() {

    int a=30, b=40;

    swapValue( a,  b);

    cout<<a<<"  "<<b<<endl;

}
```

C++ structures

- A **`struct`** is a data structure composed of simpler data types.

```
struct Point {
    double x;
    double y;
};
```

Pointers to structures

The dot operator (.) extracts a structure field.

The arrow operator (->) dereferences and extracts a structure field with a single operator.

```
struct Point {
    double x;
    double y;
};
```

```
struct Character {
    string name;
    int yearAtHogwarts;
    bool isBoyWhoLived;
    bool isCurrentlyPosessedByTomRiddlesDiary;
};
```

References to structures

Draw a diagram to show the state of memory when the function setPoint is called

```
void setPoint(Point &q, double x, double y)
{
    //Code to set the x and y values of q

}
```

```
int main(){
      Point p;
      setPoint(p, 100.0, 200);
      cout <<p.x <<“ “ <<p.y<<endl

}
```